



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

2001-03

An experimental investigation of the decision processes of software managers

Swanson, Douglas W.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/35206>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



THESIS

**AN EXPERIMENTAL INVESTIGATION OF THE
DECISION PROCESSES OF SOFTWARE MANAGERS**

by

Douglas W. Swanson

September 1995

Thesis Advisor
Thesis Co-Advisor

Kishore Sengupta
Tarek Abdel-Hamid

Approved for public release; distribution is unlimited.

19960401 025

DTIC QUALITY INSPECTED 1

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1995		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE AN EXPERIMENTAL INVESTIGATION OF THE DECISION PROCESSES OF SOFTWARE MANAGERS			5. FUNDING NUMBERS	
6. AUTHOR(S) Swanson, Douglas W.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
<p>13. ABSTRACT (maximum 200 words) In recent years, the Department of Defense has been plagued by cost overruns and schedule slippages in major software development projects. As a result, the <i>performance</i> of managers on software development projects has come under increasing scrutiny. Despite this scrutiny, little attention has been paid to the decision processes of software project managers.</p> <p>The goal of this project was to conduct micro-empirical research into managers' decision processes, thereby gaining a greater understanding of the ways in which managers make decisions. The project used a simulation of a real NASA software project and verbal protocol analysis techniques to capture the decision processes of seven professional software project managers. Methods of analysis were adapted and developed to trace and explore the captured protocols. Specifically, each subject's decision processes were traced and compared to a nominal model of the decision environment and compared to other subjects.</p> <p>It was found that, despite their experience, the subjects generally had difficulty developing mental models which accurately captured all of the complex relationships in the software management environment. The quality assurance sub-system proved especially difficult. The observed difficulties with the quality assurance decision would prove detrimental to real-world software projects and should be investigated further.</p> <p>Additionally, it was noted that managers tend to place a greater emphasis on meeting schedule goals than cost. Also, more experienced managers tend to rely on specific data, make fewer changes, and are less able or willing to verbalize their thoughts.</p>				
14. SUBJECT TERMS Software management, dynamic decision making, complex systems, verbal protocol analysis, process tracing, cognitive maps, mental models, causal links, causal relationships			15. NUMBER OF PAGES 118	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

**AN EXPERIMENTAL INVESTIGATION OF THE DECISION PROCESSES
OF SOFTWARE MANAGERS**

Douglas W. Swanson
Lieutenant Commander, United States Navy
B.S., University of Montana, 1982
M.B.A., Embry Riddle Aeronautical University, 1991

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT


from the

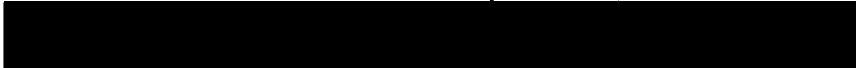
NAVAL POSTGRADUATE SCHOOL
September, 1995

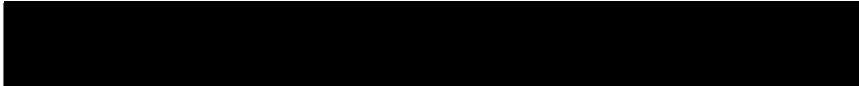
Author:


Douglas W. Swanson

Approved by:


Kishore Sengupta, Thesis Advisor


Tarek Abdel-Hamid, Thesis Co-Advisor


Reuben Harris, Chairman,
Department of Systems Management

ABSTRACT

In recent years, the Department of Defense has been plagued by cost overruns and schedule slippages in major software development projects. As a result, the *performance* of managers on software development projects has come under increasing scrutiny. Despite this scrutiny, little attention has been paid to the decision *processes* of software project managers.

The goal of this project was to conduct micro-empirical research into managers' decision processes, thereby gaining a greater understanding of the ways in which managers make decisions. The project used a simulation of a real NASA software project and verbal protocol analysis techniques to capture the decision processes of seven professional software project managers. Methods of analysis were adapted and developed to trace and explore the captured protocols. Specifically, each subject's decision processes were traced and compared to a nominal model of the decision environment and compared to other subjects.

It was found that, despite their experience, the subjects generally had difficulty developing mental models which accurately captured all of the complex relationships in the software management environment. The quality assurance sub-system proved especially difficult. The observed difficulties with the quality assurance decision would prove detrimental to real-world software projects and should be investigated further.

Additionally, it was noted that managers tend to place a greater emphasis on meeting schedule goals than cost. Also, more experienced managers tend to rely on specific data, make fewer changes, and are less able or willing to verbalize their thoughts.

TABLE OF CONTENTS

I. INTRODUCTION	1
A. DISCUSSION	1
B. PURPOSE OF RESEARCH	2
C. THESIS ORGANIZATION	3
II. THEORETICAL PREMISE	5
A. CHAPTER OVERVIEW	5
B. SOFTWARE PROJECT MANAGEMENT - A COMPLEX SYSTEM	5
C. DECISION RESEARCH USING VERBAL PROTOCOL ANALYSIS	7
D. THE NEED FOR EXPERTS	10
E. CHAPTER CONCLUSION	11
III. METHOD	13
A. CHAPTER OVERVIEW	13
B. THE EXPERIMENT	14
C. COGNITIVE MAPS AND THE NOMINAL MAP	18
D. THE CODING SCHEME	26
E. CHAPTER CONCLUSION	27
IV. RESULTS	29
A. CHAPTER OVERVIEW	29
B. SURVEY RESULTS	29
C. DECISIONS	30
D. PERFORMANCE	33
E. INDIVIDUAL PROTOCOL ANALYSIS	36
F. SUMMARY OF OBSERVATIONS	51
V. CONCLUSIONS AND RECOMMENDATIONS	55
A. CONCLUSIONS	55
B. RECOMMENDATIONS	55
APPENDIX A	57
APPENDIX B	63
APPENDIX C	67
APPENDIX D	75
LIST OF REFERENCES	105
BIBLIOGRAPHY	107
INITIAL DISTRIBUTION LIST	109

I. INTRODUCTION

A. DISCUSSION

One need not look far to see the tremendous price/performance advances in computer hardware. Software development on the other hand, has not fared as well. The literature is rife with examples of software projects gone awry. The General Accounting Office (GAO) reports that cost overruns and schedule slippages are a way of life in the Department Of Defense (DOD). Even when completed, many systems fail to meet technical requirements, and thus are of little value to the user (GAO, 1989). While efforts are under way in DOD to improve processes, management failures are still evident in DOD Information Technology acquisition. For example, a recent GAO report on the Continuous Acquisition and Lifecycle Support Initiative portrays the program "in disarray" after spending \$5 billion in the past 10 years (GAO, 1994).

The root of most problems in major Information Technology system procurements is poor software development. Frederick Brooks compares the software development process to the pre-historic tar pits that were the demise of many great beasts (Brooks, 1975). Contributing to the software management problem is an annual growth in demand of 25 per cent compared to a 4 per cent growth in the number of trained programmers (Kitfield, 1989). To overcome this disparity, managers must simply make better decisions faster. For that to happen, a greater understanding of the software project managers' decision processes and their limitations is required.

Clearly then, there is room for improvement in the software management process in both government and private sectors. This thesis presents a micro-empirical analysis of the decision making behavior of software project managers. Specifically, an experiment using verbal protocol analysis was conducted to trace the decision processes of software project managers, and methods developed to analyze the results. The experiment consisted of recording subjects as they thought out-loud while managing a simulated software project from start to finish. The recordings were then transcribed and mapped using a symbolic scheme developed for this research.

B. PURPOSE OF RESEARCH

This study explored the decision processes of software project management. By conducting this study and answering the primary research questions, a clearer understanding of how software project managers make decisions was attained. The benefit of a better understanding of any decision process is the ability to positively influence the outcome of the process. By understanding the intricacies of the decision processes involved in software management, better tools, information, measures, and systems may be developed to support the decisions. This will result in better decisions and ultimately higher quality software, developed at a lower cost.

1. Primary Research Questions

This research was intended to explore the decision making processes of software project managers. Being exploratory in nature, the exact outcome of the research was difficult to predict. However, in general, this research sought to answer the questions:

- How do experienced software project managers make decisions?
- Do software project managers' cognitive maps of the decision environment accurately portray the causal web that comprises the actual environment?
- What commonalties exist between individual managers decision processes?
- Are managers consistent in their own decision processes?

2. Scope of Research

Using an existing simulation based on the Systems Dynamic Model (SDM) of Software Project Management (Swett, 1995; Abdel-Hamid and Madnick, 1990), an experiment was designed and executed to extract the decision processes of experienced software project managers using verbal protocol analysis. Methods to analyze those protocols were developed to answer the primary research questions.

3. Limitations

Protocol analysis is an extremely time intensive research method. As such, the primary limit to this research was time. Therefore, some methods of analysis are developed here, but their full implementation is left for future studies.

C. THESIS ORGANIZATION

The thesis is organized into five chapters. Chapter I provides an overview of the purpose and scope of the research. In Chapter II, a brief review of pertinent literature is discussed. A working background in decision theory and software project management is developed. Chapter III provides a detailed discussion of the design and execution of the experiment. The methods of analysis are then discussed. Chapter IV details the results of the analysis described in the previous chapter. Chapter V will present a summary of the findings and recommendations for future research.

II. THEORETICAL PREMISE

A. CHAPTER OVERVIEW

Decisions are often made difficult by the complex, dynamic systems or environments in which they are made. One such environment envelops software project managers. This research project is intended to explore how software project managers make decisions in the complex system of software development. Towards that end, this chapter will first demonstrate the complexity of the software management environment. This is accomplished by fitting the characteristics of an existing model of the software development environment to characteristics of a general complex system. The last section will provide a review of verbal protocol analysis as a means of conducting decision research. The need for expert subjects in decision research is also presented.

B. SOFTWARE PROJECT MANAGEMENT - A COMPLEX SYSTEM

Software management is a complex system. While this is self-evident, it is worth exploring the specific characteristics of software project management which make it complex. This section will present characteristics of a complex system in a general sense, then match those with specific characteristics of software project management.

Funke (1991) defines complex problem solving or decision making in terms of several criteria. Briefly, he characterizes problem solving in complex systems by the following:

- *Intransparency* - Not all variables are observable. Frequently, only symptoms are apparent.
- *Polytely* - Multiple and conflicting goals exist for the decision maker.
- *Connectivity of variables* - Changes in one variable change many other variables, making it difficult to conceptualize all consequences.
- *Dynamic developments* - Spontaneous changes occur in the situation or system, requiring immediate action by the problem solver.
- *Time delayed effects* - Effects of certain actions are frequently delayed. This is in direct contradiction to the immediate action required by system dynamics.

Recognizing the need for micro-empirical research in the field of software project management, a series of studies have been conducted on the effects of various environmental factors on the decisions made by software managers (Hardebeck, 1991; Abdel-Hamid, 1992;

Abdel-Hamid et. al., 1993; Baker, 1992; Bosley, 1994; Swett, 1995). These experiments were all conducted using graduate students as subjects, and a simulation interface and model built in Dynamo. The simulation model is a mathematical representation of the Systems Dynamics Model (SDM) of software project management (Abdel-Hamid and Madnick 1989, 1990).

Abdel-Hamid and Madnick's model accounts for literally hundreds of variables and their complex interrelationships. In the following bullets, examples from the SDM will be cited to demonstrate how the various characteristics of complex systems presented by Funke (1991) apply to the software management environment.

- *Intransparency* - While many of the hundreds of variables in the software management system are frequently invisible to the manager simply because of the number of variables in the system, perhaps nowhere is this more evident than in the QA sector (Abdel-Hamid and Madnick 1990). While a manager should be aware of the number of defects discovered, it is no surprise that the actual number of defects will never be known. Likewise, the manager will never know the actual amount of the project completed at any given time, or the exact amount of communication overhead affecting productivity.
- *Polytely* - Multiple conflicting goals are the very heart of any management problem, especially the software management problem. Swett (1995) demonstrated the conflicting nature of two simple sets of goals: minimize cost overruns vs. minimize schedule overruns, and minimize schedule overruns vs. maximize quality. When adding in the plethora of other goals facing the project manager, the complexity is magnified exponentially.
- *Connectivity of variables* - A review of the various sectors of Abdel-Hamid's model will leave little doubt regarding the connected nature of the variables in this process. Hiring delay illustrates the point clearly. In the Human Resource Management Subsystem (Abdel-Hamid and Madnick 1990), hiring delays dictate the newly hired workforce and therefore the total workforce. Total workforce then influences manpower available for QA and development in the Manpower allocation sector, as well as directly impacting productivity in the Productivity Sub-sector. Meanwhile, in the Planning Subsystem, willingness to change workforce is effected and thus the actual workforce level sought. When compounded by assimilation delays, which account for the time a new hire needs to become savvy on the given project, the effects are nearly incomprehensible.
- *Time delayed effects* - Two of the highest impact time delay effects have already been discussed: hiring delay and assimilation delay. Others include the delay in incorporating newly discovered tasks, delays in adjusting job size, and exhaustion depletion delay time. (Abdel-Hamid and Madnick 1990)

- *Dynamic developments* - Dynamic developments address the tendency of the project to spontaneously change. Again this characterizes the many unforeseen developments in any management project, but is most pronounced in the software realm in the area of requirements. The literature is rife with examples of projects where requirements doubled during the development cycle. Indeed, the exception would be the project in which requirements did not change.

Sterman (1994) further elaborates on the impact of time delays on the decision process. He also notes that dynamic complexities not only limit the ability of decision makers to effectively cope in a dynamic system, but also impede learning by causing misperceptions of feedback. By misinterpreting feedback, decision makers reinforce dysfunctional behavior, and fail to learn from their mistakes.

C. DECISION RESEARCH USING VERBAL PROTOCOL ANALYSIS

This section will discuss some fundamentals of decision research and review literature pertinent to the analysis used in this research.

1. Decision Research

The field of decision research is extremely complex and widely diverse. Available literature is so extensive that a comprehensive review is infeasible. However, a partial review of the literature consulted in preparation of this experiment is appropriate. Specifically, that body of work dealing with verbal protocol analysis will be explored.

Much of decision research involving protocol analysis is based on Ericsson and Simon's book, *Protocol Analysis - Verbal Reports as Data* (1984). Not only did this text summarize the work to date, it described detailed methods for collecting, reporting, and analyzing protocols. In short, it laid the foundation for current protocol analysis techniques. The methods developed by Ericsson and Simon were used extensively in this research, particularly in the execution of the experiment.

Verbal protocol analysis involves the extraction and analysis of a subject's mental processes by recording a verbal description of the process. The description can be given either retrospectively or concurrently. While a detailed discussion of the two will not be provided here, concurrent verbalization was determined to be more appropriate for this research. Retrospective reporting depends more on unreliable long-term memory and has been shown to be more problematic on a number of fronts. It is important to note that collection of concurrent protocols

must be done carefully so as not to bias the subject's response. Basic instructions should be provided up front (see Chapter III), and only neutral comments by the researcher made to encourage verbalization during the actual experiment (Ericsson and Simon, 1984; Carroll and Johnson, 1990).

In conducting protocol analysis, Svenson (1989) stressed the importance of a sound reference model. In her research, Svenson distinguishes between primitives used in the representation of the task and the coding scheme. She asserts that the reference model, or task representation, is a symbolic description of another system, with primitives being the smallest units in the system. A scheme for coding verbal protocols can be developed, based on that representation. The smallest units of the coding scheme may or may not be the same as the task primitives. This insight was used to develop the nominal model of the software management decision task environment. Svenson states:

...Most verbal protocols actually contain a lot about the information processed and very little or nothing about how it is processed. Therefore it is important to have a good theory or model which makes it possible to understand the data in a process-oriented language. Very rarely do protocols themselves provide the theory for their understanding. Instead, the abundance of information tends to overwhelm the unsophisticated researcher, making it very difficult to find a structure for an understanding or explanation of the process that produced so much data.

Providing even more specific guidance, Carroll and Johnson (1990) describe the steps in representing the task or, in their terms, completing the task analysis.

It starts with the surface features of the decision by considering the decision maker, the alternatives to be chosen or judgments to be made, the information available to the decision maker, and the circumstances of the decision... The task analysis then goes deeper into the processes that are likely to occur as the decision is made. First, decision makers have a representation or mental image of the decision or judgment task, called a *problem space*, formed through the interaction of task information with prior knowledge. The researcher tries to specify the possible states of knowledge or contents of short-term memory that the decision maker might have. Second, the researcher identifies a set of goals and sub-goals for the decision, which correspond to the desired locations or directions in the problem space. Third, the researcher tries to specify a set of *operators* that act on the contents of short-term memory to create new knowledge or -- in our terminology -- move the decision maker from one point in the problem space to another....

Carroll and Johnson's strategy was essential in developing the nominal model of the task environment for this research (see Chapter III). The purpose of the nominal model is to provide a basis for comparison of subjects' decision processes. Decision processes are traced from the verbal protocols. The resulting process trace represents the subjects mental model or cognitive map of the decision environment.

Sterman (1994) describes the role of mental models and how 'misperceptions of feedback' adversely affect peoples mental models. He further argues that human cognitive limitations make it difficult at best for people to 'simulate mentally even the simplest possible feedback system, the first-order linear feedback loop', and virtually impossible to grasp high-order, non-linear relationships, even when they are supplied with perfect knowledge of the system. This distinction between linear and non-linear relationships is key to our analysis of peoples cognitive maps, and how these models relate to the real world causal structure, which will be represented by our nominal model. For the purpose of this research, we will define linear relationships as causal chains, and non-linear relationships as causal webs.

2. Analyzing Decision Protocols

There are seemingly as many methods for analyzing protocols as there are experiments using them. Viewing protocol analysis as one type of process tracing, one review examined 45 studies using various approaches to process tracing (Ford, et. al., 1989). Carroll and Johnson (1990) distinguish three categories of analysis: informal exploratory analysis, content analysis of statement frequencies, and construction of formal simulation models. This research will employ methods fitting the first two categories.

The exploratory analysis involves simply transcribing and reading the protocol and making observations. It is most appropriate when only limited information exists about the task. While this may seem rudimentary, a large amount of information can be gained from exploratory analysis. Which attributes are used, the methods employed for combining attributes into judgments and decisions, and differences between subjects' decision processes (Carroll and Johnson, 1990) can all be discerned from observation. The analysis in this research goes slightly beyond mere observation, and compares a high level trace of each subject's processes to the task representation, yielding an even richer understanding of the decision processes.

Content analysis is more rigorous and intensive than exploratory analysis. It involves developing a coding scheme, segmenting the transcribed protocol into single complete thoughts, and finally coding each segment. Once coded, a number of analyses can be performed on the protocol; frequency, sequence, and transition to name a few (Carroll and Johnson, 1990). A coding scheme was developed for this experiment, but due to the nature of the research and time constraints, its use will be deferred to follow on research.

While these categories of analysis are useful in a general sense, specific examples of techniques used in similar experiments provide further guidance in developing a method of analysis for this particular research. One study used verbal protocols to examine the option generation process and employed a number of different analysis techniques, including frequency and state/transition diagrams. In that study, the coding scheme was based on the SHOR (Stimulus-Hypothesis-Option-Response) paradigm. This paradigm recognizes decision makers as "problem solvers who gather information, generate causal hypotheses, and generate and evaluate options and take action on the basis of these hypotheses". Once coded, analysis of variance, frequency analysis, and state-transition analysis were conducted on the coded protocols. While the coding scheme did not translate directly to this experiment, it did provide a framework on which to base a coding scheme. The frequency and state-transition analysis were also useful. (Adelmann, et. al., 1995)

Another experiment used verbal protocols and the *UCLA Executive Decision Game* to explore strategic business decision making. This study was similar in complexity to the decision tasks and environment found in software project management. This coding scheme provided valuable insight into coding causal relationships. The notion of differentiating light causal analysis from deep causal analysis was especially useful. (Schweiger, et. al., 1985)

D. THE NEED FOR EXPERTS

As previously noted, past experiments have used graduate students with training in software management as subjects (Hardebeck, 1991; Abdel-Hamid, 1992; Abdel-Hamid et. al., 1993; Baker, 1992; Bosley, 1994; Swett, 1995). Given that these experiments focused largely the results of the decision makers given different changes to the environment, this was acceptable. This research focuses on the process of decision makers, rather than the performance. Studies show that the decision processes used by experts are quite unique.

Camerer and Johnson (1991) note that experts use contingent search, search less, and use more knowledge in their decision making. Contingent searches refer to the behavior of experts in considering different sets of variables in different sequences, depending on the particular case.

The special characteristics of the decision processes of experts make it desirable to use them in process research. For this reason, this research used seven professional software managers as subjects.

E. CHAPTER CONCLUSION

In reviewing the literature, it is clear that software management is a difficult, complex process. While much has been done to explore the results of various filters on the decision makers *performance*, little research to date has sought to expose the *processes* decision makers use. What's more, no studies of experienced project managers were noted. Verbal protocol analysis is a method available to begin investigating these processes, and shows some promise in understanding the way managers view the decision problem. While a variety of techniques have been used to analyze verbal protocols, each research situation requires a unique approach to relate the protocols to the research questions.

III. METHOD

A. CHAPTER OVERVIEW

As stated, the purpose of this research was to gain a greater understanding of how software project managers make decisions. Towards that end, an experiment was designed to extract the protocols of ten professional software project managers, trace their decision processes, and analyze the results. Only seven of the ten subjects protocols were analyzed for reasons to be discussed later in the paper.

The protocol extraction was accomplished by video taping each subject as they thought out loud while making decisions in a simulated software management environment. The audio portions of the recordings were transcribed for analysis. The video portions provided a useful reference for establishing context and for clarifying subjects' remarks when the audio was unclear. The simulated environment utilized a previously developed experimental interface to the Systems Dynamics Model (Swett, 1995). To allow subjects to become familiar with thinking out loud and with the interface, subjects were given practice sessions in verbalization and in using the interface. During the simulation, the values of each variable and each decision were extracted for later analysis. Following the simulation, each subject responded to a questionnaire in order to gain some additional information on demographics and feedback on the experiment.

In order to trace the subjects' decisions, it was first necessary to develop a reference model of the decision environment. This model is referred to as a nominal map throughout this paper. The nominal map provided a basis for comparing the cognitive maps of the subjects. By reviewing the transcribed protocols, cognitive maps of each subject's decision process for each 40 day period were developed. To facilitate a more rigorous analysis, an formal coding scheme was also developed. To provide a transcribed protocol to use in the development of the coding scheme, the experiment was administered to a trial subject. Due to time constraints, this coding scheme was developed but never used in this research. The scheme is presented here so that it may be used in future research.

The actual analysis consisted of several steps. First, the final outcomes of individual experts were plotted. Individual outcomes were used to support general observations regarding the demographics of the group of experts and their performance. Next, the four decision

variables of each subject was plotted for each period. Again these were used to identify general trends or phenomenon and to relate them to the demographics when possible. Finally, the maps of each subject were compared to each other and to the nominal map, and the results summarized.

This Chapter will discuss the conduct of the experiment and the development of both the nominal map and the coding scheme. The analysis and results are presented in detail in Chapter IV.

B. THE EXPERIMENT

In this experiment, subjects were required to manage an actual software project. The project profiled an actual NASA software project in which requirements grew over 50% during the course of the project. Management of the project consisted of making decisions on how large their total staff should be and what percentage of total staff should be allocated to QA. In addition, they were required to update their estimates for total project cost and duration (schedule). Decisions were based on information presented in six available screens. The screens consisted of three tables and three graphs. The tables presented status information, such as current requirements, reported progress, and reported productivity; staffing information including actual staff and experience level; and QA information including defect density and the QA expenditure rate. The graphs provided trends on selected variables from the three tables. Decisions were to be made at intervals of 40 working days. Swett (1995) provides a complete description of the interface and simulation environment. In addition to the information presented, subjects were given a specific goal of minimizing cost and schedule overruns. The requirements profile and goals matched one of four groups of students in the Swett experiment. This provided the basis for an expert versus student comparison.

Prior to conducting the final simulation, it was necessary to allow subjects to practice thinking out loud and practice using the simulation interface. Both of these aspects of the experiment, as well as subject selection are discussed below.

1. Subject Selection

Ten subjects were initially chosen to conduct the actual experiment. They were selected based on their availability and experience. All ten subjects worked as software development managers in a medium size, scientific data collection and research facility. After being given

guidelines for subject qualifications, the facility selected subjects and relayed the names to the researchers.

To reveal any bias in the background of the subjects, a detailed survey was developed and administered following the completion of the experiment. The survey and results are presented in Appendix A. A summary of the survey results pertaining to experience is shown in Table 1. Based on the results of the survey, the protocols of subjects 2 and 4 were not used in the analysis, since they had no experience in software project management, and failed to qualify as experts.

Subject Number	Age	Years Managing Software Projects	Type of Software Project Management Experience	Education
1	over 55	18	Semi-detached; FORTRAN, BASIC, C/C++; Mainframe, PC, Workstation	BS(Edu)/ MS(Metro)
2	41-45	Development Only	No management, development only	AA/(CS)
3	over 55	23	Semi-detached; FORTRAN; Mainframe, Super-Computer	BS/MS/ PrePhD
4	under 30	0	None	BS(PhySci)/ MS(Ocean)
5	41-45	7	Organic; C, FORTRAN, Assembly; PC, Workstation. Embedded; Assembly; Mainframe	BS(HR/CS)
6	over 55	5	Embedded; FORTRAN, Ada: Mainframe, Workstation	BS/MS(Phy)/ MS(Met)
7	41-45	12	Semi-detached; FORTRAN; Super-Computer	BS(Ocean)/ MS(Metro)
8	46-50	17	Semi-detached; FORTRAN, Assembly, C; Super-Computer, Mainframe, Workstation	BS(Eng)
9	46-50	3	Semi-detached; FORTRAN, Ada, C; Workstation	BS(Geol)/ MS(CompSci)/ MS(Metro)
10	51-55	10	Embedded; FORTRAN; Mainframe, Workstation	BS(Geol)/ MBA

Table 1. Subject Experience

2. Practice Verbalization

Ericsson and Simon (1984, pp. 377-8) provide a sample of general instructions and practice problems to be used when gathering think-aloud reports. The general instructions and first two practice problems used in this experiment were adapted for the particular task. They were as follows:

In this experiment, we are interested in learning what you think about when you make decisions at each step of the simulation. In order to do this, I am going to ask you to THINK ALOUD as you work on the problem. What I mean by think aloud is that I want you to tell me EVERYTHING that you are thinking from the time you start the simulation until it is complete. I would like you to talk aloud CONSTANTLY through the entire simulation. I don't want you to try to plan out what to say or try to explain to me what you are saying. Just act as if you are alone in the room speaking to yourself. It is most important that you keep talking. If you are silent for any long period of time, I will ask you to talk. Do you understand what I want you to do?

Good, now we will begin with some practice problems. First, I want you to multiply these two numbers in your head and tell me what you are thinking as you get an answer.

"What is the result of multiplying 24×36 "

Good. Now I will give you two more practice problems before we proceed with the experiment. I want you to do the same thing for each one of these problems. As before, I want you to think aloud as you think about the question. Any questions? Now, here is your next problem.

"How many windows are there in your parents house?"

These two practice problems were followed with a more in depth problem designed to allow the subjects to recognize and vocalize deeper causal relationships in preparation for those found in this experiment. The third problem posed was, "In a market-based economy, expectations of inflation lead to actual inflation. Please explain how." In the event subjects were not familiar with the domain or were otherwise unable to form a response, cues were given to get them started. Figure 1 shows the relationships the subjects were expected to vocalize.

Ericsson and Simon (1984) point out that these practice problems help subjects verbalize concurrently with problem solving.

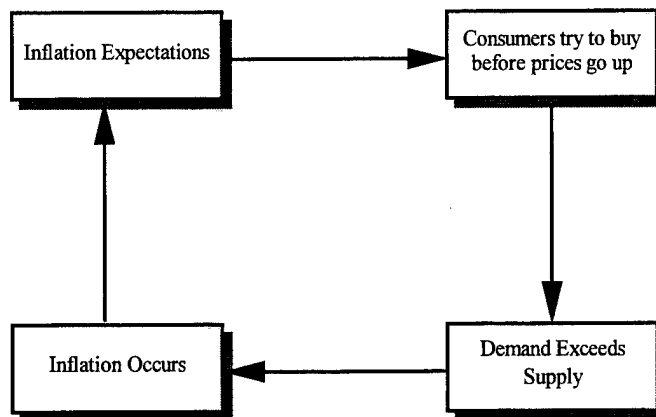


Figure 1. A Causal Model of Inflation

3. Practice with the Interface

Past experiments with this type of interface have indicated that the interface can be an impediment to understanding the decision problem (Bosley, 1994; Baker, 1992; et. al.). To overcome this, a simulation of a generic management problem was provided, with no explicit goals, and no unexpected changes in requirements to deal with. The interface for this practice problem is identical to that of the final simulation. During the practice simulation, subjects are permitted to ask questions regarding the interface and presentation of information on the screen. Techniques for analyzing the data and strategies for making decisions are not open for discussion.

4. The Final Simulation

All parts of the experiment were administered over a one week period to ten subjects. Each subject was scheduled for two two-hour time blocks, not more than two days apart. During the first block, they were given both the practice verbalization and the practice simulation. No data was collected and nothing recorded during this session.

During the second block, the subjects were administered the final experiment. They were asked to vocalize throughout the entire simulation and were recorded as they did so. When necessary, subjects were prompted to encourage thinking aloud. As suggested by Ericsson and Simon (1984), these prompts were very neutral, such as, "What are you thinking now?".

To ease the burden of transcribing the protocols, a high resolution 8mm tape was used for video and audio recording. Playback equipment allowed replay directly on a multimedia PC. As

the session was replayed, the subject's thoughts were translated verbatim to text. The text was then segmented into single topic thoughts or statements. These statements were then used to develop the coding scheme as described later in the Chapter.

Prior to both the practice and the final simulation, the subjects were given written instructions which told them specifically what to do and gave them initial estimates for the project. The instructions for the final simulation are included in Appendix B. The practice instructions varied only in the initial estimates and the goal.

C. COGNITIVE MAPS AND THE NOMINAL MAP

1. The Subjects' Cognitive Maps

After all subjects completed the simulation, the protocols were transcribed. Once the transcriptions were complete, each subject's decision processes for each period were mapped graphically using the symbology described in Figure 2. The symbology is comparable to that used on the nominal map. For consistency, elements that are common to multiple maps are located in the same place on each map. In this way, the maps may easily be compared. The process used to generate the individual cognitive maps for each subject will be clarified in Chapter IV.

2. Developing the Nominal Map

The importance of having a representation of the decision task was emphasized in Chapter II. As discussed, Abdel-Hamid and Madnick's (1990) Systems Dynamics Model (SDM) for software project management was used as the basis for this simulation. As such, it also provided the basis for the nominal map. By pairing elements of the SDM with the symbology used in the subject maps, the nominal map was developed using the same symbology as the subject maps, but with the causal relationships of the Abdel-Hamid and Madnick model, which has been extensively verified. This yielded a nominal map which could then be used to compare and evaluate the subjects' maps. Chapter IV discusses the results of this comparison, and the comparison of the subjects' maps to each other.

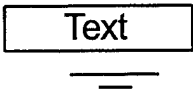






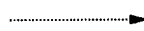



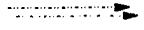
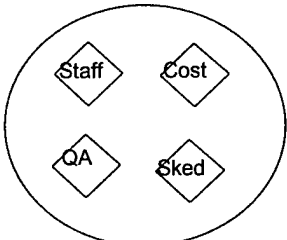
	Information from a source, such as a report or a graph.
	Information calculated or derived by the subject
	Subject made an assumption, applied a heuristic, or otherwise brought in outside decision aids.
	Addition operator. May also be multiplication, division, or subtraction.
	Evaluation. Indicates an evaluation or comparison. Evaluation of only one item indicates evaluation with past values for that item.
	Justification. Indicates the subject engaged in some sort of justification of a past decision.
	Explicit information flow. The subject specified the use of information.
	Implicit flow. The use of the information was implied, but not specified.
	Accurate link. The subject accurately identified a causal link between the linked items.
	Inaccurate link. The subject identified a link but it was inconsistent with the nominal map.
	Partially accurate link. The subject identified a link which captured part but not all of a causal relationship.
	Miracle or Blackhole. Subject either made a decision not traceable to any information flow or pursued an information flow which was never used in any decision making, either explicitly or implicitly.
	Decision group. This is the center of the map for each period. It contains the cost, schedule, staff, and QA decisions. The outer ring groups the four decisions. Flows may originate or terminate on either the ring or the decisions, depending on how specifically the subject tied the information flow to a given decision.

Figure 2. Map Symbology

Obviously, capturing all the complex elements of the SDM was not only impractical but undesirable. The subjects couldn't be expected to incorporate all causal linkages in their cognitive maps given the information they were provided. Therefore, the nominal map is a much higher level look at the decision environment than provided in the SDM. It concentrates on three areas of the software management decision environment that are crucial to this simulation, namely, the Staffing decision, the Quality Assurance decision, and estimating Cost and Schedule. For the purposes of discussion, each of these three areas is described and illustrated as a separate segment in the following sections.

a. The Staffing Decision

Determining the staffing level is one of the key decisions a manager must make, yet it involves many factors. While not a complete list, the following factors are discussed in Abdel-Hamid and Madnick (1990, Chap. 5).

- The current perceived tasks left to be completed and the amount of scheduled time remaining must be factored into the staffing decision.
- Hiring delays result in the size of current staff lagging behind requested staff.
- New hires are not immediately productive. There are delays in training and communications overhead to consider. Knowledge of these assimilation delays effect the willingness of the manager to change the workforce.
- Implicit limits on the number of new hires that can be absorbed at a time and on the total workforce exist and must be considered. These limits are also embodied in the manager's willingness to change the workforce. They reflect the notion that drastic changes to the staffing level will adversely effect productivity.
- The stability of the workforce must be considered. Stability refers to an assessment of how long new hires will be needed. Stability changes depending on the stage of project completion. An example of consideration of stability would be the reluctance to add new hires late in a project, choosing instead to increase schedule.
- There is constant turnover in any project that must be considered when assessing staffing level.

The Staffing Decision map in Figure 3 captures these factors graphically. The map basically shows that all of these factors must be evaluated when making a staffing decision.

Given the above discussion, the map is for the most part, self explanatory. As in the SDM, the notion of a managers willingness to change the workforce captures a number of the factors discussed. It is also worth noting that the 'Resources Required' block is a function of the perceived tasks left to be completed and perceived productivity. This encompasses several relationships that will be explored further in the Cost and Schedule Estimation map. The fact that the decision maps overlap in the figures illustrates the difficulty in separating the decisions from each other.

Staffing Decision

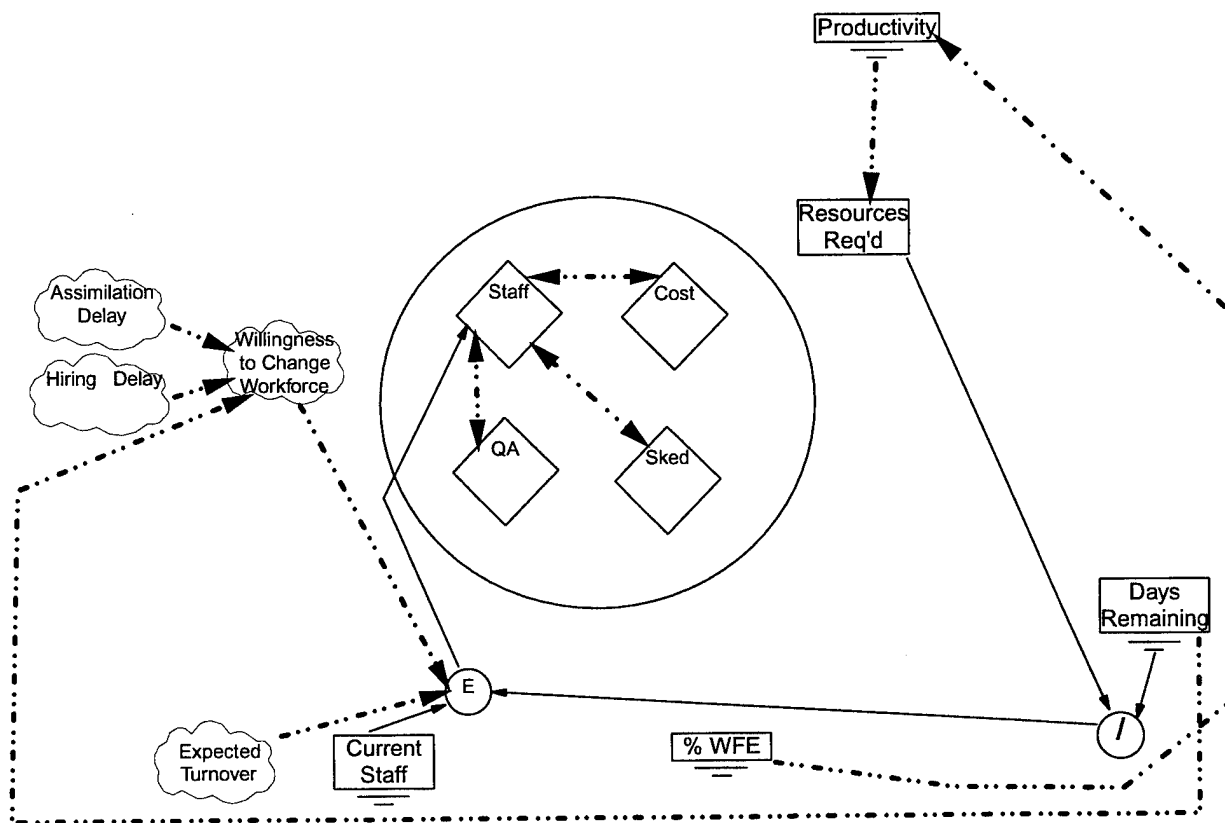


Figure 3. The Staffing Decision

In addition to the factors discussed, the map illustrates the relationships between the staffing decision and the other three decision variables. An increase in staffing level may increase or decrease cost and schedule depending on the other factors in the system. Conversely, budget or schedule constraints will effect the staffing decision. The QA system may be affected

because a large workforce may generate a higher density of defects, due to increased communications overhead. The negative effect of increased staff size on productivity is also modeled, as is the link from experience to productivity.

b. The Quality Assurance Decision

The Quality Assurance (QA) subsystem is both interesting and complex. It is made so largely because of the simple fact that the actual number of defects in a real project can never be known with certainty. In the face of that uncertainty, managers are left with the task of optimizing their staff allocated to the QA effort. Chapters 8 and 18 of Abdel-Hamid and Madnick (1990) provide a thorough discussion of the QA system and the associated economics. Several salient points are:

- A number of studies have confirmed a significant savings when defects are discovered early. It is also evident that the error generation rate declines as a project progresses, partly due to an increasingly experienced workforce.
- The optimal QA effort varies between projects, and is also dependent on the environment.
- It is impossible to detect all defects, and, depending on the environment, a QA effort beyond a certain percentage of total development effort yields diminishing returns.
- Increasing QA effort as a percentage of total development effort results in an *exponential* increase in QA's actual cost in person-days. Since beyond a certain point, increases in QA person-days no longer result in corresponding decreases in rework and testing, the total cost of the project increases. This leads to a larger work force (or schedule delays). A larger work-force results in lessened productivity.

Figure 4 shows a map of these QA relationships. The map graphically represents the points about the QA system discussed above. It shows that in attempting to determine an appropriate level, the defect density during a period can be balanced against the QA person days per thousand delivered source instructions (kdsi). This can be accomplished in the context of the simulation interface by monitoring the graphs showing trends in defect density and QA person-days per kdsi. For example, a declining density with no change or an increase in QA expenditures per kdsi may indicate a reduction in QA manning level is appropriate. Evaluating density and expenditure rate with past periods data, defect expectations, and with a predefined

QA plan which recognizes the benefits of early detection, will provide a basis for adjusting QA level as the project progresses. Consideration must also be given to workforce experience, since it effects error generation rate. All of these elements are evaluated together resulting in the QA allocation decision. That decision in turn effects total staff, cost, schedule, and productivity as noted in the above discussion.

QA Decision

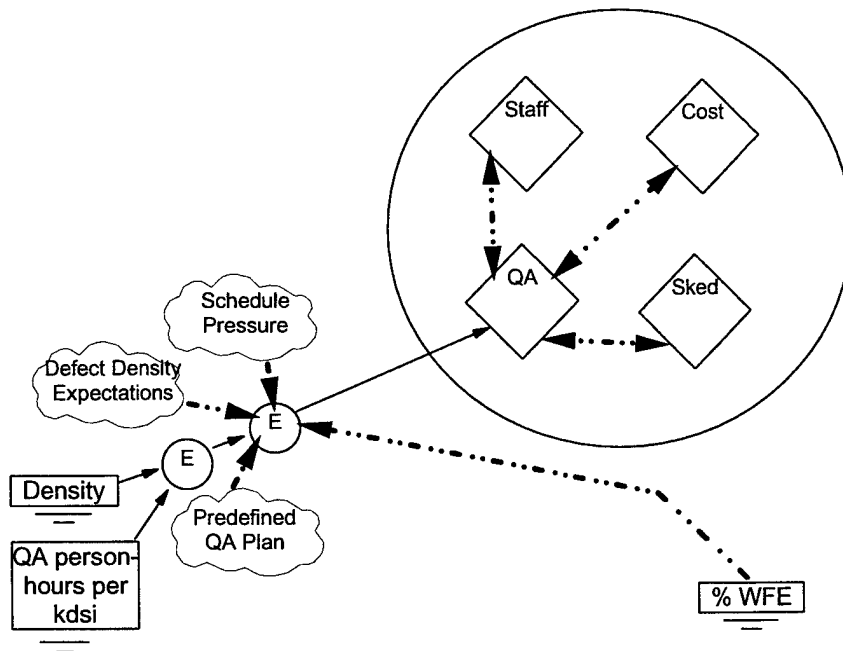


Figure 4. The QA Decision

c. Cost and Schedule Estimation

Updating estimates of total programming cost and scheduled completion time are two additional key duties to be performed by the manager. They are presented together because the two estimates are so intertwined as to be practically inseparable. According to Abdel-Hamid and Madnick (1990, Chapters 10 and 11), the updating of cost and schedule estimates involves elements of both the controlling and planning subsystems. The controlling subsystem involves measuring progress and expended resources on the project. The planning subsystem

encompasses the initial project estimates and their updates through the projects life. The following factors are noteworthy:

- Any control system consists of at least three elements: *Measurement*, or detection of what is happening; *Evaluation* which usually involves a comparison of what is happening to some expectations; and *Communication* or report of what is happening so it can be used.
- It is difficult to measure progress in software development. Usually in early stages of a project, reported progress is based on resources expended. As the project progresses, accomplishments become more visible and reported progress and productivity become more a function of perceived tasks completed.
- When adjustments to cost are perceived necessary, managers first try to absorb them via staff and productivity. Only when this fails are they willing to adjust the projected cost of the project.
- The schedule determination can be made by simply dividing the cost in person days by the number of persons on the project to obtain the number of days required for project completion.
- Schedule pressure can have a significant effect on productivity. When a project is perceived to be behind schedule, developers are capable of bursts of productivity for short periods of time. (Abdel-Hamid and Madnick, 1990, Chapter 7)

The map in Figure 5 illustrates one way which the cost and schedule estimates might be achieved. It shows the decision process starting by calculating requirements remaining and dividing by perceived productivity to get a new estimate of resources required to complete the project. Resources expended must be added in to obtain the new cost estimate. This in turn can be divided by staff to obtain the days required for the total project. In this view of the decision problem, it is assumed that the staffing decision is made first. Staff will be adjusted if appropriate considering the willingness to change workforce and other factors modeled in the Staffing Decision. The notion that schedule pressure affects productivity is also modeled. This effect is observable in the simulation, but only after the fact. In other words, subjects may see a seemingly inexplicable productivity burst resulting in a large jump in reported progress as they approach their estimated duration.

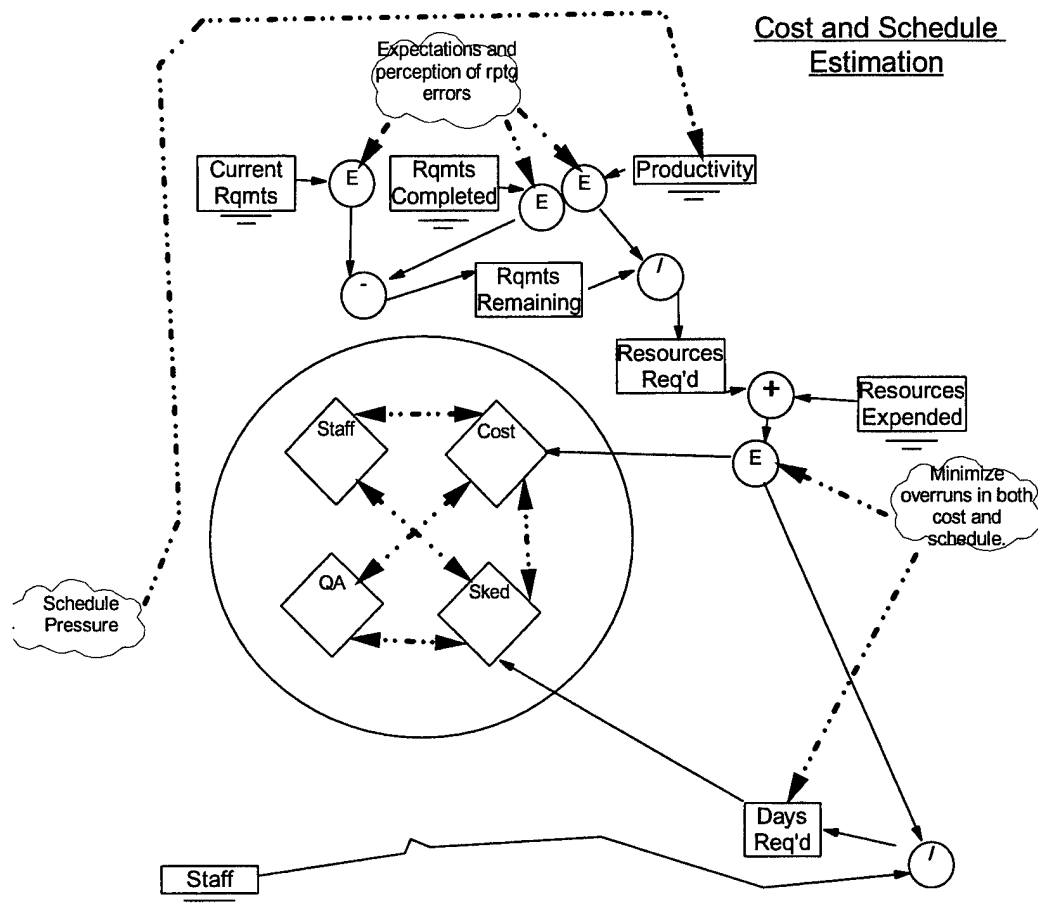


Figure 5. Cost and Schedule Estimation

At some point in the estimation process, a decision maker should acknowledge the inherent unreliability of the reported progress on which the decisions are based, and expectations of future changes in requirements and productivity. This notion is specifically modeled for productivity, requirements, and requirements completed. It also applies to percent reported complete, which is simply another way of viewing reported progress. If a manager has any expectations of future changes in requirements or productivity, or if they perceive any errors in reporting, they adjust the figures accordingly. Also, in the case of our experiment, the subjects were given a specific goal of minimizing overruns in both schedule and cost, so they could reasonably be expected to factor in a tradeoff between the two. Both of these relationships are

modeled in the map. Again, the non-linear links between the four decision variables are represented. These have already been discussed in the previous two sections.

d. The Complete Nominal Map

After having analyzed each of the three areas, they can easily be combined to form the nominal map of the decision making environment for this experiment. The complete map is shown in Figure 6.

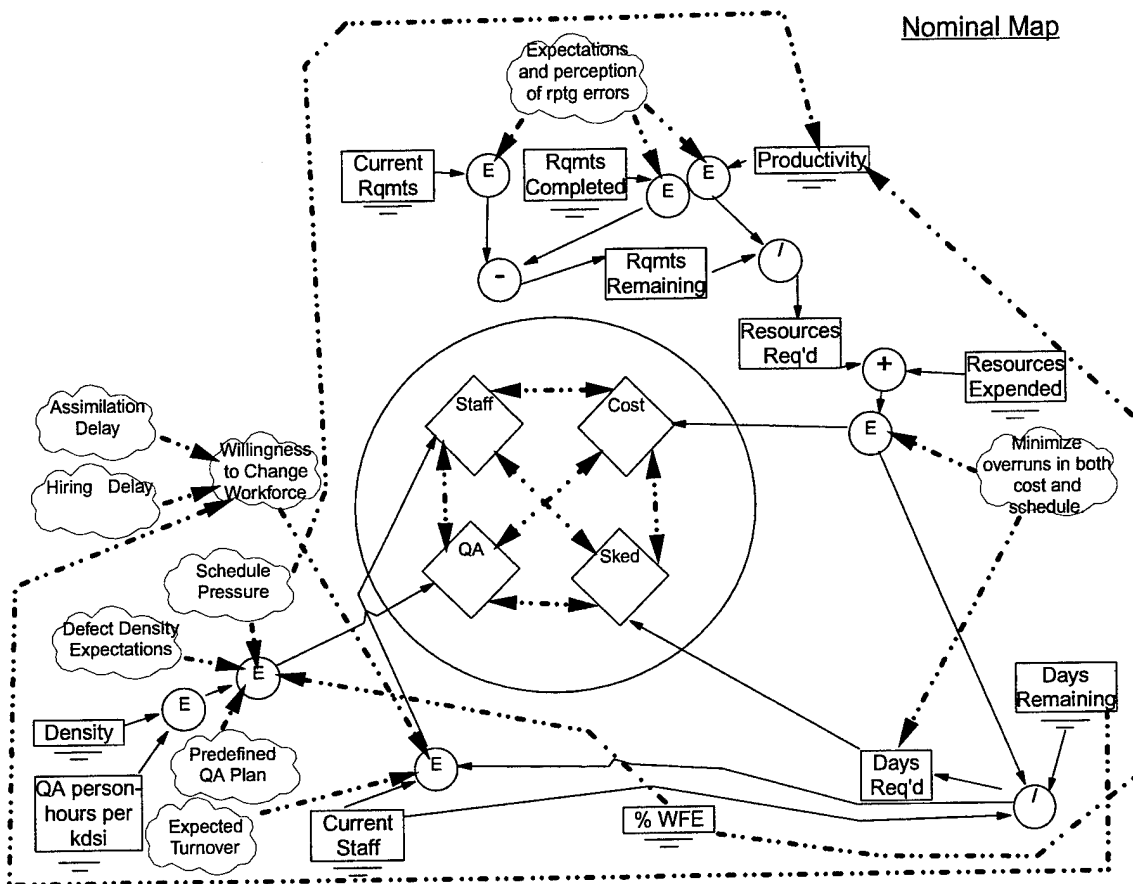


Figure 6. The Nominal Map

D. THE CODING SCHEME

While the mapping scheme provides the basis for discussion of the subjects cognitive maps, a more rigorous coding scheme is needed to analyze the protocols thoroughly. Such a scheme was developed, but time constraints did not allow its' use. A complete description of the scheme is presented in Appendix C.

E. CHAPTER CONCLUSION

In this Chapter, the experimental design and the methods used to capture the protocols were discussed. The methods developed to structure the resulting data were also discussed. This included the development of a mapping scheme and a detailed coding scheme.

Chapter IV will present the results of mapping each subjects decision process and comparing those maps to a nominal map and to each other.

IV. RESULTS

A. CHAPTER OVERVIEW

In this chapter, the results of the analysis discussed in Chapter III will be presented. First, the interesting aspects of the survey results will be discussed. Next, the decisions for all subjects during each period are plotted and analyzed. The performance of each subject's simulation will then be explored by looking at the outcomes of their decisions. Finally, the maps for each subject's decision processes will be discussed by period. By analyzing the survey responses, the decisions, the performance, and the maps of each subject, a complete picture of that subject's mental model of the decision problem will be evident.

It is important to note that while ten subjects participated, only the processes of seven are analyzed. As previously discussed, two subjects were dropped because they had no program management experience. A third subject was dropped when it became apparent that he had never recognized that the project was growing. Consequently, his decision processes were not comparable to the others. The subjects not used were numbered 2, 4, and 5.

B. SURVEY RESULTS

The survey and a summary of the results are presented in Appendix A. Several interesting points are worth noting:

- Of the two goals of minimizing overruns in both cost and schedule, 6 of the subjects felt they placed a greater emphasis on schedule than cost. The one remaining subject, subject 3, felt he gave the two goals equal consideration.
- Five of the subjects felt the reports were more useful than the graphs in making decisions. The remaining two, subjects 3 and 9, rated them equally useful.
- Software project management experience of the group ranged from 5 years to 23 years. Subjects 1, 3, and 8 all had over 17 years while the rest were under twelve. The three subjects over 17 were observed to have a more difficult time verbalizing during their decision processes. This was especially true of subjects 1 and 3.

C. DECISIONS

The simulation required each subject to decide on an initial total staff and a percentage of total staff to allocate to QA. During each subsequent 40 day increment, the subjects were also required to estimate cost and schedule. Each of the decisions for each subject by period were extracted during the simulation and are plotted in Figures 7 - 10. Each of the decision graphs will be discussed in turn. Each subjects decisions will be discussed in greater detail in Section E of this chapter.

1. Staffing

The general shape of the staffing curves (Figure 7) followed the growth profile of the project. In other words, the subjects tended to add people as they perceived the project was growing. For instance, subject 8 recognized the requirements growth immediately, and began adding in the third period. Most of the subjects tended to be reluctant to add people late in the project. In fact, no one requested additional people within 3 periods of the projects completion.

Several subjects deserve special attention. Subject 7 assumed a 50% growth in the project and also elected to commit a high percentage of staff to QA. This explains his high initial staffing decision. Subject 9 also exhibited interesting behavior when he reduced staff early in the project. This was done prior to recognition of the requirements growth, and was intended to improve project performance by getting rid of the inexperienced workforce. The subject quickly hired additional people when he recognized the requirements growth. Subject 8 seemed to overreact to the requirements growth, and subsequently reduced staff at the end of the project.

2. Quality Assurance

The quality assurance decisions are plotted in Figure 8. The behavior here was not as consistent as the staffing decision. Ideally, errors are detected early on in the project, reducing the cost of rework. In practice however, only three subjects behaved according to this notion. Subject 7 expressly chose a QA profile which started high and tapered off towards the end of the project. Unfortunately, while the shape of this profile was consistent with the idea of early detection, the percentages throughout were much higher than they needed to be. Subject 8 also voiced a desire to detect defects early, but was overcome with the requirements growth towards

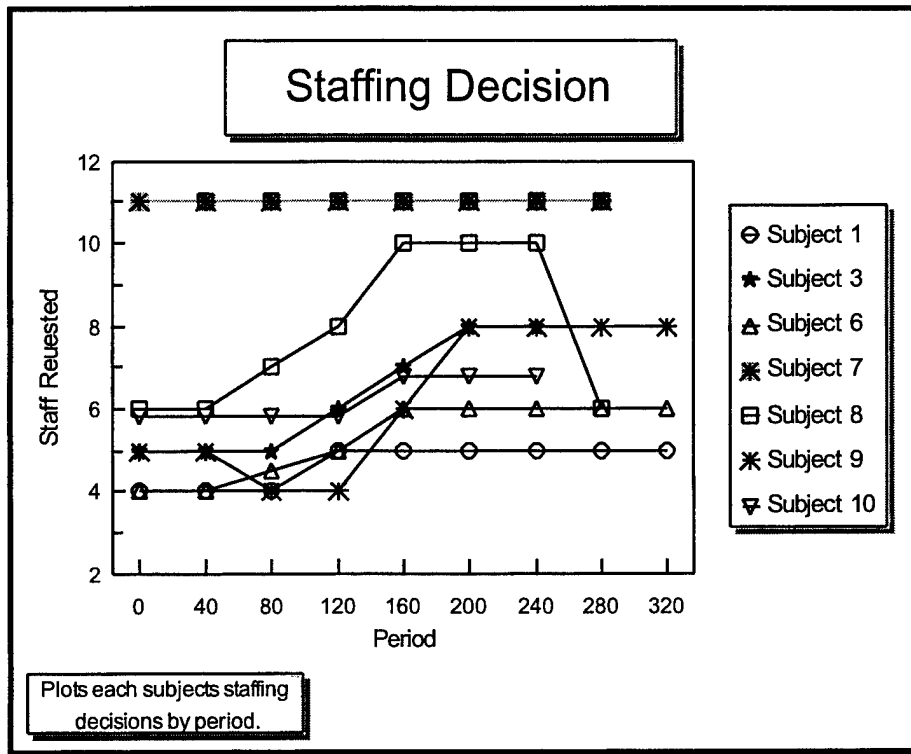


Figure 7. The Staffing Decision

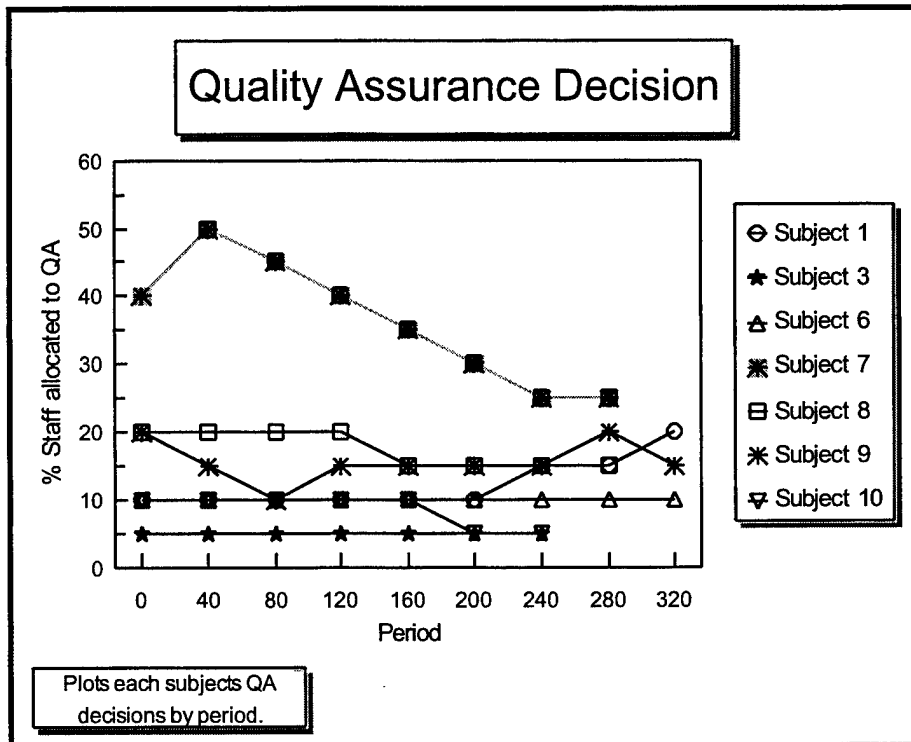


Figure 8. The QA Decision

the end of the project. Subject 9 also expressed a desire to detect defects early. During the simulation, he adjusted the QA level in an effort to keep the defect density at a target level. Towards the end of the project he recognized that defect density would fall off regardless of effort, since there simply were not as many defects to detect. While sub-optimal, this proved to be one of the more effective strategies.

Subjects 3, 6, and 10, seemed to have no firm strategy for QA. They either accepted the default of 10% or reduced the level even further to improve productivity. Subject 1 clearly stated he believed QA early in the project was a waste of time. He demonstrated this by increasing QA level over the course of the project.

3. Cost

Figure 9 shows the subjects cost estimates plotted over time. As one might expect, the subject's estimated cost increased as the project grew. The exceptions to this were subjects 10 and 3. These two subjects never increased their cost estimates even when resources expended exceeded the estimate. This indicates a possible failure to grasp this facet of the project. Subject 7 had a high curve simply because of the higher staffing level used.

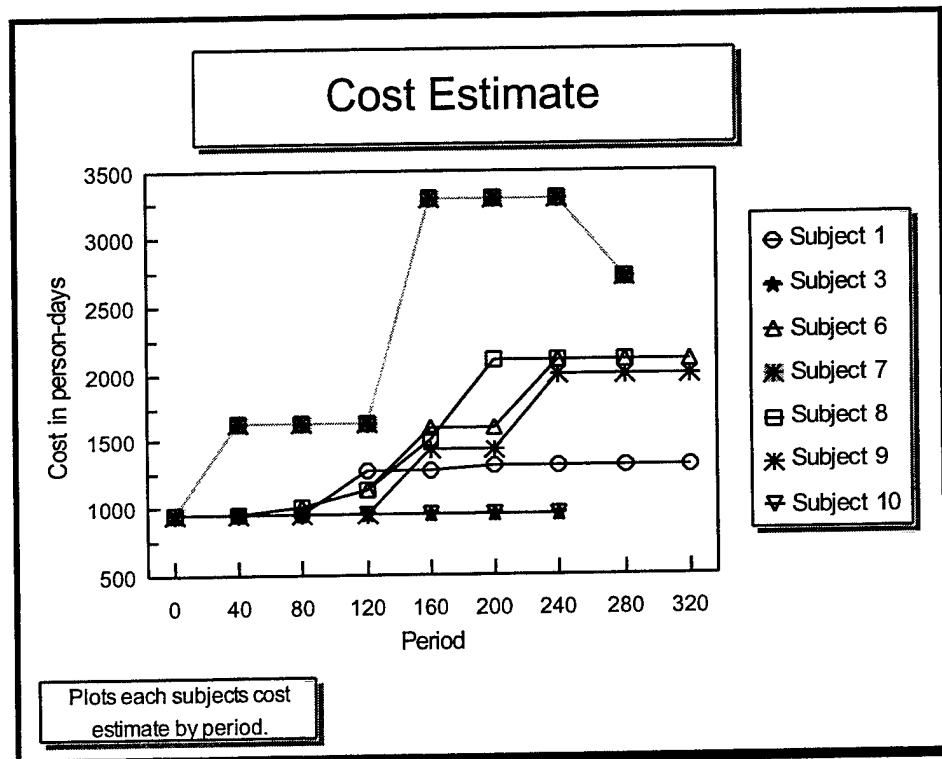


Figure 9. The Cost Estimate

4. Schedule

Figure 10 shows the schedule estimates of the subjects over time. Like the cost estimates, these curves generally followed the growth profile of the project. Subject 3 again made no changes, although he actually did complete the project within the allotted time. Subject 9 made an errant calculation which drove his estimate high for two periods. Interestingly, the subjects seemed much less willing to increase their schedule estimates than the cost estimates. Many of the cost estimates doubled over the course of the project, while the schedule estimates grew by less than a third.

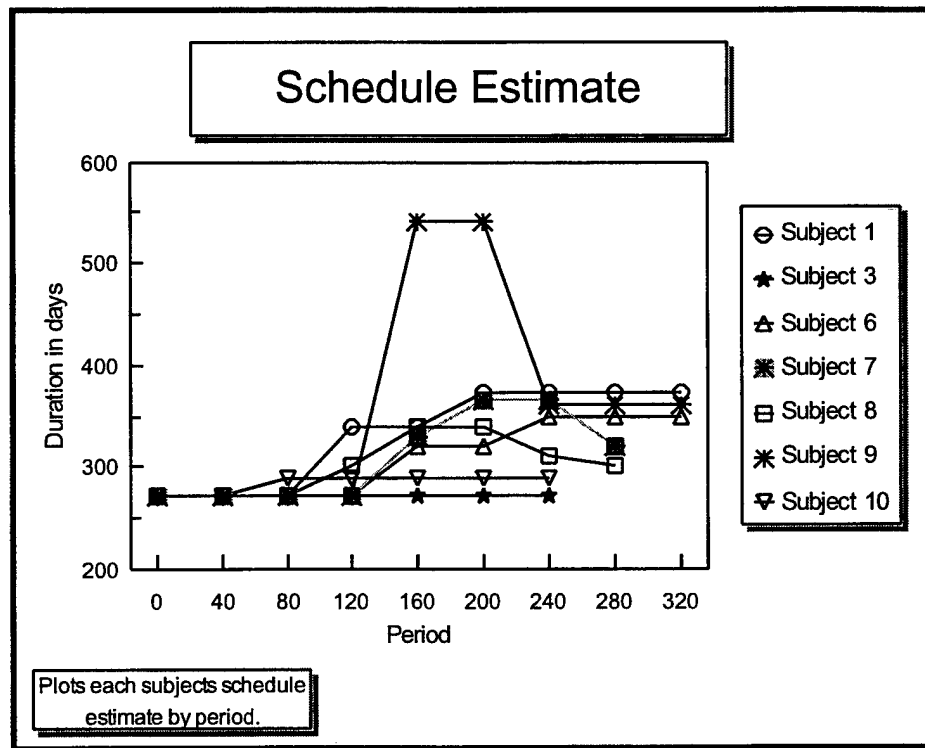


Figure 10. The Schedule Estimate

D. PERFORMANCE

While the thrust of this investigation was to explore the decision processes of managers, a brief look at their performance is also useful. The performance of each subject can be measured in terms of defects in the project, total cost and total duration. These data points are extracted during the simulation and plotted in Figures 12-14.

1. Defects

Looking at the total defects in each subject's project can provide some interesting insight into the impact of their decisions. As shown in Figure 8, Subject 3 paid little attention to defects over the course of the project, allocating only 5% of staff to QA. Figure 11 shows this resulted in over 6000 defects - three times the next highest subject. Subject 10 allocated only 10% and cut defects to roughly 2000. On the other end of the spectrum, subject 7 allocated as much as 50% of staff to QA, but defects were only marginally less than subjects 8 and 9 who never allocated more than 20%.

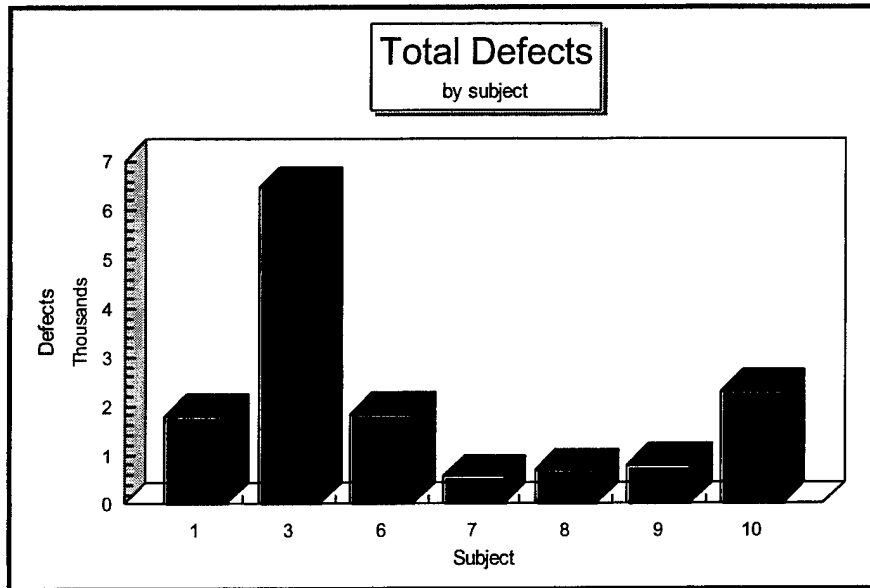


Figure 11.

2. Final Cost

Comparing the project cost to QA and total defects provides further insight into the consequences of the staffing and QA decisions. Figure 12 shows that subject 7 clearly paid a premium for a marginal improvement in quality. Comparing subjects 3 and 10, we see that subject 3 saved only a few person-days by ignoring QA, while subject 10 improved quality by a factor of three for almost no additional man-days. While QA is not the only factor contributing to project cost, Figures 11 and 12 demonstrate graphically how failing to optimize QA may result in either a poor quality product or an exorbitant cost.

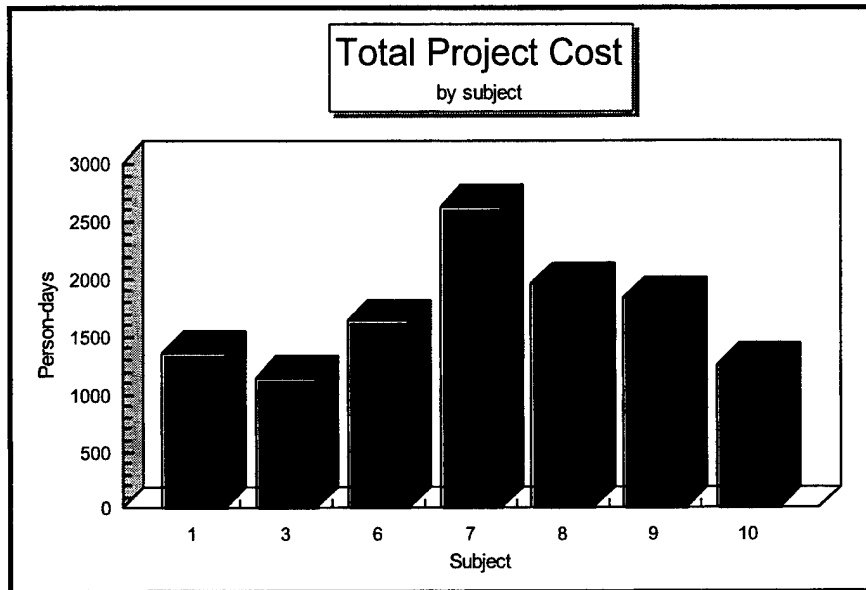


Figure 12.

3. Duration

Figure 13 shows the final duration of each project. The outcomes in terms of duration do not vary as dramatically as the final costs. This is consistent with the priorities subjects placed on meeting their scheduling goals. While cost and schedule can be traded off to some degree, most subjects placed a greater emphasis on meeting their schedule goal.

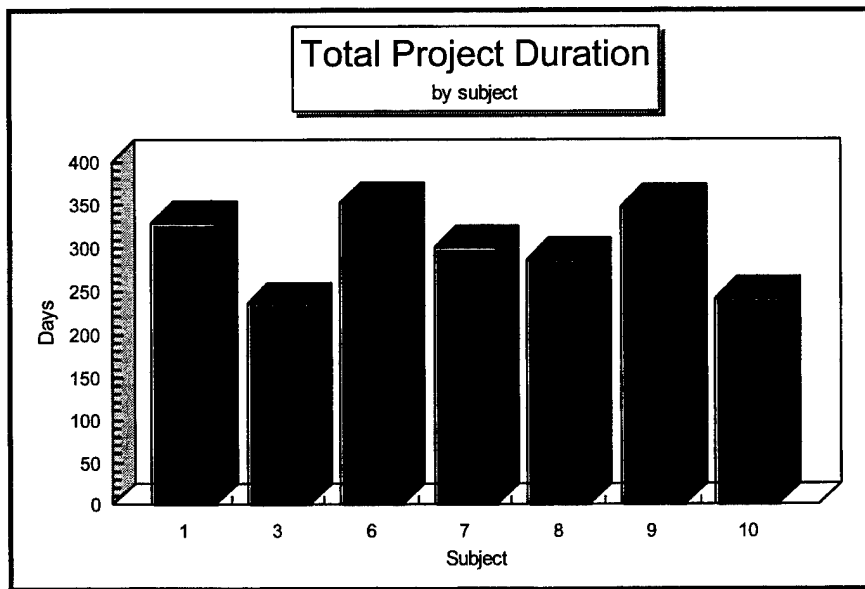


Figure 13.

E. INDIVIDUAL PROTOCOL ANALYSIS

1. Mapping the decision processes

As discussed, the verbal protocols for each subject were analyzed after being transcribed. The mapping scheme developed in Chapter III was used to develop a map of each subjects decision process for each period. The maps are contained in Appendix D. The symbology used is the same that was used in the nominal map. Figure 2 in Chapter III provides a legend of that symbology.

To provide amplification to the graphic view, a narrative for each of the subjects is included here. The narratives are broken out by period, and elaborate on interesting aspects of the subjects' maps, especially heuristics used and links developed. These narratives are intended to be used to compliment the maps in Appendix D.

a. Subject 1 Narrative

Initial decision. The subject computed the underlying productivity assumption for the initial estimates, but did not explicitly use this information in any decisions. He then divided cost by schedule to determine initial staff size and rounded up to the next whole number. The initial QA level was based on the suggested 10% and a heuristic voiced by the subject which demonstrated the belief that QA should be done primarily towards the end of the programming phase. Based on that heuristic, the subject used a predefined profile for QA level.

40 days. Only staff, work force experienced, and defect data were viewed by the subject. The defect density compared favorably to the subject's expectations. The subject did not notice the increase in requirements.

80 days. During this period, the subject began to compare reported progress to elapsed time. The apparent discrepancy was then correctly linked to the observed level of experience. This link captures the concept that a less experienced workforce will be less productive. This notion is illustrated in the nominal model as the link between workforce experience and productivity. The increasing requirements were recognized during this period, and implicitly tied to staff and schedule. The subject felt that in response to the increasing requirements, he may have to increase staff or schedule. This simple link between staff and

schedule did not demonstrate a grasp of the complexities of the web existing between the four decision variables (Figure 7).

120 days. Here, resources required to complete the project were computed, then added to resources expended to obtain a new estimate for total cost. This was then divided by the old cost estimate to obtain a factor increase of 1.3. The factor increase was then applied to the schedule estimate as well. This demonstrated an incorrect linear link between cost and schedule. Having accounted for increasing requirements by updating the cost and schedule estimates, the subject then inexplicably added an additional person to the staff, apparently just for good measure. This is demonstrated by the flow originating on the decision group ring and terminating on the staff decision.

160 days. Having increased the staff by one last period, the subject felt his decision was justified this period by another increase in requirements. The subject then compared the requirements growth with their expectations for normal growth. Through this comparison the subject determined that the initial planning was inadequate and that in the "real-world" they would call some meetings to try and get the project under control.

200 days. In this period, the subject seemed to become slightly overwhelmed by the complexity of the problem. This is evidenced by the dead end flows and the dysfunctional update to cost. First, the subject computed a factor increase in requirements from the beginning of the project, apparently to apply to cost and schedule, but then never used the information. He then computed the resources required to complete the project, but rather than add this to resources expended as at 120 days, he compared it to resources remaining for no apparent reason. The subject then correctly used the resources required to compute additional time required, but erroneously added the 32 days to both cost and schedule, clearly a mistake. Comments voiced by the subject did illustrate a reluctance to add people late in the project. The subject also demonstrated a simple link between staff and the other decision variables, although a complete grasp of the interrelationships was still not evident.

240 days. The subject noted a slowing of the requirements increase and an improvement in the teams productivity. Based on the reported progress, the subject applied his predefined QA plan and increased the staff allocated to QA.

280 days. There were no changes during this period. The subject merely reviewed some of the information available, and commented that the graphs were not very useful.

320 days. Again in this period, the subject reviewed a few data points and applied his predefined profile for QA staffing and increased QA effort.

Summary. This subject demonstrated a rudimentary grasp of some of the relationships. He seemed to understand the staffing decision reasonably well, however the relationships between cost and schedule eluded him. While he had a QA plan, it did not account for the potential savings from early detection of errors, nor did he assess his QA performance based on cost per error.

b. Subject 3 Narrative

Initial decision. The subject calculated the productivity assumption underlying the initial figures. He also divided requirements by schedule, but never voiced the result. He then calculated an initial staff and adjusted it up based on an expected .65 'efficiency'. His notion of efficiency is captured in the model in productivity expectations (Figure 6). The subject commented that he felt QA at the beginning of the programming phase was a waste of time. He then elected to allocate 5% of staff to QA.

40 days. The subject did a poor job of verbalizing, but essentially just reviewed the reports and graphs. He recognized the requirements increase immediately, and appeared to assess progress versus elapsed time, but made no changes.

80 days. During this period the subject was more thorough in his review of information. Although he noted the requirements increase, he generally felt that there were insufficient changes in the project to alter any of his decisions. He stated that the requirements increase might justify 'another quarter person or so', but that he would let it go for another period. He also expressed doubt about the accuracy of reported data, specifically percent reported complete, consistent with the nominal map (Figure 6).

120 days. The subject quickly scanned the reports but keyed on the requirements growth and the reported progress versus elapsed time. He accurately identified a tradeoff between extending the project or hiring more people (Figure 4). He also inaccurately extended

this notion to include a direct tradeoff between cost and staff. He seemed to feel that by increasing staff, he avoided cost overruns, even in the face of increasing requirements. In the end, he chose to increase staff to 6 and make no other changes.

160 days. Since this subject did not vocalize very well, it was difficult to trace the entire decision process. Some interesting points were observable however. The subject was extremely frustrated with the increasing requirements, and remarked that in a normal project, that level of requirements growth would be grounds for halting the project. Despite the requirements growth, he was much more willing to increase staff than update the schedule or cost estimates, despite the pressure of the approaching deadline. As a result, he held cost and schedule constant, but increased staff to 7. This was apparently in consideration of his goals, which he reiterated during the period. He also commented that QA was 'not my problem', referring again to the goals of minimizing overruns in cost and schedule. This was apparently further justification for his relatively low allocation to QA, but it ignores the fact that insufficient QA effort can result in excessive rework, which can in turn lead to cost and schedule overruns.

200 days. The subject made many of the same assertions as in the previous period, and again opted to increase manning based on the increasing requirements. In this instance, there was no evidence of an assessment of progress, merely a response of increasing staff by one based on increasing requirements. He did note that experience had dropped down, but failed to recognize this was linked to his increases in staff. He also did a comparison of defect density and QA expenditures per kdsi, but incorrectly concluded that since the defect density was rising and expenditure rate was falling, his QA effort was acceptable. In reality, the fact that he was detecting more defects with lower expenditures per kdsi was a good indication that there were an increasing number of defects in the code.

Summary. While this subjects thoughts were not as clearly voiced as some others, several points were evident. First the subject placed a heavy emphasis on meeting his goals. As a result, he held constant the cost and schedule estimates while steadily increasing staff in the face of increasing requirements. In doing so, he failed to recognize the cost associated with bringing in staff late in the programming phase. Second, he failed to note the cost of rework associated with an insufficient percentage of staff allocated to QA. He further misinterpreted the defect density and QA expenditures to support his QA allocation decision, when it really

indicated a higher QA level might be appropriate. Finally, the interrelationships between the decision variables were never clearly established. He was very frustrated by the increasing requirements which undoubtedly compounded his difficulty in seeing the relationships in all of these areas.

c. Subject 6 Narrative

Initial decision. Like several of the subjects, this subject calculated the underlying productivity assumptions of the initial estimates, but then never explicitly used that information. He based his initial staffing decision solely on the estimated cost divided by schedule. The QA allocation was made strictly on the basis of the suggested value.

40 days. During this period, the subject merely browsed the reports and graphs, then opted to leave the decisions unchanged.

80 days. The subject recognized the increasing requirements immediately in this period. He responded by computing a percent increase in requirements, then applying that as a linear factor to both staff and cost. This approach seems logical but neglects the actual productivity of the programmers. It makes no allowance for the progress made on the original requirements. The subject compared the trends in defects detected and QA expenditures, and opted to leave QA manning the same. Schedule also remained unchanged.

120 days. Much like the previous period, the subject applied a linear factor to the staff requested and the cost estimate, based on the increased requirements. He did note a decline in productivity that he accurately attributed to his growing staff (Figure 4). He made no change to QA or schedule, but did comment that he was 'running out of time'.

160 days. During this period the subject eventually demonstrated a fairly thorough grasp of the cost and schedule estimation aspects of the decision problem. He identified the increase in requirements, and compared his progress to elapsed time. Initially, he reacted to the new increase in requirements by trying to increase staff, cost and schedule by the same factor as requirements. He eventually used productivity and remaining requirements to estimate resources required, then rounded this up. By dividing this by a new staff of six, he came up with a new estimate for duration. His decision to increase the staff by one more person was

based on the amount he had allocated to QA, expected turnover, and to some degree on the increase in requirements.

200 days. The subject again noted progress lagging behind elapsed time, even with the new schedule estimate. He re-estimated cost and found it to be close to his previous estimate, primarily because he had rounded cost up the previous period. Based on that and a review of the other data, he opted to leave all estimates unchanged for another period. It is noteworthy that he considered increasing staff, but was reluctant to do so. The exact cause for the unwillingness to change wasn't clear.

240 days. In this period, the subject recognized that requirements had increased again, then re-estimated cost based on the new requirements. Finding his previous estimate still sufficient, he calculated that it would take an additional 1.5 people to finish the project in the time remaining, but was unwilling to request the additional staff because of the hiring and assimilation delay and the limited time remaining. Based on his current staff, he then calculated an additional 29 days would be required to complete the project. Had he stopped there, his process would have been fairly accurate, unfortunately he then multiplied his new estimate for days by the current staff to get a new cost estimate. Since the current staff was much higher than average staff during the course of the project, this resulted in an erroneously high cost estimate, clearly a mistake. This indicated only a partial understanding of the relationships between cost and staff. He also compared defect density to QA expenditures, and found density rising with falling expenditures. Like subject 3, he misinterpreted this to justify his current allocation QA, when in reality, it probably indicated that a growing number of defects were evident in the code, and were therefore easier to detect.

280 days. The subject computed the resources required and used them to check his schedule estimate, but never checked his estimate for cost. He also compared progress to elapsed time to evaluate his schedule estimate. He then noted a decline in the defect density, and accurately attributed it to the improving experience level of the workforce (Figure 5). He also compared the trends in cost growth with the requirements growth, which should have been an indicator of the error in his cost estimate. He was satisfied with the status and made no changes to any of the decision variables, having failed to catch his excessive cost estimate.

320 days. Based on the days remaining and the resources required to complete the project, the subject recognized that he could reduce staff, but was unwilling to do so. He again failed to recognize his previous error in the cost estimate, even after seeing costs climbing relative to requirements. He attributed the fact that requirements had grown by 50% while costs had doubled solely to assimilation delay. In reality, his cost estimate was merely in error.

Summary. This subject demonstrated a fairly robust understanding of the problem space. He correctly linked staff, cost and schedule, with the exception of his cost estimation error. He accurately expressed the impact of hiring and assimilation delays, and how they effect the staffing decision late in the project. Like many of the subjects, a complete grasp of the QA problem eluded him. Also like many subjects, misinterpretations tended to justify the subjects past decisions. This is evident both in his evaluation of defect density versus QA expenditures, and his evaluation of cost versus requirements growth.

d. Subject 7 Narrative

Initial decision. This subject had a very well defined approach to the problem. His staffing decision consisted of first making the initial estimate two different ways, then applying a correction factor of three to each estimate, and then intuitively choosing between the two. The first estimate was made by simply dividing cost by schedule. The second estimate involved dividing requirements by duration, then dividing by an assumed productivity of 8 lines of code per person day to get staff. The adjustment factor of three consisted of 1.5 for potential increases in requirements, and 2.0 as a general rule of thumb. Both estimates came out close (10.5 and 10.9) and he rounded up to 11 for an initial staffing decision. His initial QA level was determined by a predefined profile in which he planned to start out high on the first period, go higher still in the second, then slowly taper off the QA effort in the latter stages of the project. While this plan accurately allowed for the economies of uncovering errors early, it did not recognize the increased costs of placing too great an emphasis on QA.

40 days. The subject evaluated the percentage complete versus the percentage of time used, the actual productivity versus his expectations, and looked at staff and experience level. By looking at current staff versus expected staff, the subject estimated an average staff over the project, then removed the previous allowance for 50% project growth, then multiplied

by total duration to develop an estimate of project cost based on initial requirements. After looking at the defect density, he increased his QA effort to 50 % based on his predefined profile. These events indicated at least a partial understanding of the linkages between staff and cost, but still failed to demonstrate the linkage between QA and cost and the notion of diminishing returns on QA effort. The subject did state the density was 'pretty high' indicating a comparison with some defect expectations.

80 days. The subject looked at much the same data as during the last period. He adjusted QA down to 45% based on the predefined profile, and linked QA level to productivity. Since he still didn't recognize the negative effect of over manning QA, this link to productivity was deemed partially accurate. He again commented that defects were 'awful high'. The subject also accurately linked experience to defect density and to productivity. That is, as experience goes up, defects should decline, and productivity should rise. Based on the expected improvements in productivity, the subject felt the disparity between elapsed time and task completion was cause for concern but not alarm. The subject did comment at this point that he did not find the graphs of the information useful and did not intend to use them. Had the subject viewed the graphs, he may have identified the increasing requirements sooner, and factored that into his evaluation.

120 days. The subject voiced a growing concern over the continuing slide in productivity. He failed to conclude that this was due to the excessive QA level and correspondingly high staff. The subject continued to follow his predefined profile regarding QA level, lowering QA to 40%. Other decisions remained the same.

160 days. The subject noted that productivity was beginning to improve, but the progress did not compare favorably with elapsed time. He looked closely at experience, staff, and defects. He felt that the defect density was still high enough to justify his slow reduction of QA, rather than a faster reduction. Based on this, he opted to reduce the QA level another 5%. At this point, he recognized an increase in requirements, but also noted that his initial staffing estimate had allowed for it. He calculated the resources required to complete the project and, based on the days remaining, estimated that it would take a staff of 16 to complete the project. The subject felt this was not within acceptable limits for total staff, so chose to extend the project instead, demonstrating an unwillingness to change the workforce. Using the calculated resources

required to complete the project, he divided by the staff to determine additional days required to complete, which he then applied to the schedule estimate. He also updated cost, but inexplicably used his new estimate for total duration times the current staff to get the total cost of the project. Had he merely added the resources required to those expended, he would have had a more accurate cost estimate. Aside from this apparent lapse, the subject demonstrated a fairly complete understanding of the cost/schedule/staff relationship (Figures 4 and 6). He had still not recognized the impact of the QA level on the other variables.

200 days. The subject again compared reported progress to elapsed time. He then corrected his previous error in estimating cost by taking days remaining times staff and adding it to resources expended to obtain the total cost. He then recognized the new requirements and repeated the process, re-estimating both cost and schedule. He again commented that his initial expectations had allowed for a 50% growth. Interestingly, the subject also considered future improvements in productivity, but opted for the more conservative reported figures. The decision process for this period demonstrated a fairly thorough understanding of the cost and schedule estimation problem. After looking at defects, the subject continued his predefined profile for QA.

240 days. The subject first noted the addition of more requirements, but stressed that it was still within his initial allowance for growth. He was pleased that progress was now even with elapsed time and productivity was on the rise. He noted an improved experience level and declining defects. The subject felt that he was poised for very good performance in the upcoming period. The only change made was to continue to decrease the QA effort, lowering it to 25%.

280 days. Recognizing a small additional increase, the subject re-estimated cost and schedule, similar to approaches in previous periods. He noted that he had a choice between shortening the schedule or reducing staff based on improving performance. After considering his initial goals of minimizing overruns in both cost and schedule, he expressed a reluctance to reduce staff, and chose to reduce duration.

Summary. This subject demonstrated a thorough grasp of the entire problem space with only one notable exception. He failed to account for the diminishing returns in QA staffing level. By allocating an extraordinarily high percentage of staff to QA, he forced his total

staff up, resulting in excess communications overhead, and subsequently decreasing productivity. The productivity was also directly and negatively affected simply from having a smaller percentage of the staff generating code. The net result was a much costlier project completed behind schedule.

e. Subject 8 Narrative

Initial decision. The subject computed the productivity assumption underlying the initial requirements and cost estimates. He also computed an initial staff based on the initial cost and schedule estimates. His experience led him to believe that actual productivity would be lower, and he expected some delays in 'bringing people on board'. These assumptions or heuristics were the basis for raising the initial staff from 3.5 to six. Based on his experience, he set the initial QA allocation to 20%.

40 days. The subject immediately noted the change in requirements, although he felt that it wasn't significant at this point. He compared the percent of the project completed with the percentage of person days used. He noted that productivity and defect density compared favorably to what he would expect. No changes were made to any of the decisions.

80 days. The subject again recognized the increase in requirements and calculated the additional cost required. For some reason, he mistakenly used the initial productivity of 17 dsi per person day instead of the reported productivity to make the new cost estimate. He also only calculated the additional incremental cost associated with the new requirements. This in effect assumes that performance to date has gone as expected and will continue to do so through the initial requirements. As a result of these two errors, he only raised estimated person days required to 1000. He decided to add an additional person to deal with the increasing requirements, but hold schedule constant, based on the increasing requirements. While the staffing decision wasn't based on a specific calculation, and despite his other oversights, he did display a partial understanding of the cost, schedule and staff relationship (Figures 4 and 6). He elected to leave the QA allocation level unchanged.

120 days. The subject recognized an increase in requirements, and a lag in progress compared to expenditures. This period, the subject used his expectations of productivity to calculate additional resources required. He reasoned that the reported

productivity was reasonable based on workforce experience and could be expected to come up as experience level rose. He then used his current staff to calculate additional days to complete the project. While he used a more appropriate figure for productivity, he again did an incremental cost estimate, ignoring performance on the initial requirements. Finally, the subject reluctantly elected to add one more person to the staff based on expectations that requirements growth would continue. The subject seemed to understand the cost/schedule and staff/schedule relationships, but it wasn't clear that he understood the additional costs incurred from staffing changes. QA remained unchanged.

160 days. This period, the subject's decision process was almost identical to the last. Based on incremental requirements, cost and schedule were updated. Then the subject increased the staff level by two, basically stating the requirements increase outweighed concerns over adding people this late in the project. This decision seemed to indicate that he did not fully appreciate the staff/cost relationship. He did note declining defect density and elected to decrease his QA level to 15%, consistent with a QA plan which recognized the benefits of early detection.

200 days. Expressing expectations that requirements would start to level off and productivity would improve, the subject re-estimated cost, this time based on all requirements remaining, rather than just the incremental increase in requirements. He found no increase in schedule necessary, and made no changes to either staff or QA. He did state that in retrospect, he would have tried to build staff up sooner to anticipate the requirements changes, rather than bring them in over the course of the project. These comments indicated a better understanding of the cost/schedule relationship (Figure 6) than was evident from his previous decisions.

240 days. During this period, the subject left cost the same, but re-estimated duration and opted to decrease the schedule. He considered reducing staff, but didn't 'trust the changes in requirements'. His comments regarding staff indicated at least a partial understanding of the cost/staff relationship (Figure 4). The subject accurately attributed a decline in both the defect density and the QA expenditures to workforce experience. He elected to leave QA unchanged at this point.

280 days. The subject was very surprised at the progress made during this period. He noted that at 280 days, he wasn't that far over on schedule, but was way over on cost. This

was not in agreement with his initial goal of minimizing overruns in both cost and schedule. Based on that observation and the remaining requirements (1000 dsi), he reduced staff by four and reduced the schedule to 300. He felt that this would only make a marginal difference in cost, so cost remained unchanged. This analysis was evidence of a fairly complete understanding of the cost/schedule/staff relationship. QA also remained unchanged.

Summary. Over the course of the experiment, this subject demonstrated a thorough grasp of the cost/schedule/staff relationship. His analysis of the QA problem was promising, but he never fully recognized the opportunity to reduce QA allocation and save costs as the defect density declined towards the end of the project. Initially, he had alluded to a profile which called for early detection and reduced QA effort towards the end. It seemed in practice however, the QA problem took a back seat to the cost/schedule/staff decision instead of being part of it.

f. Subject 9 Narrative

Initial decision. After reviewing the initial estimates, the subject made an initial staffing decision of 5, but was not clear what he based that decision on. He specifically reiterated his goals of minimizing overruns in cost and schedule, but again it wasn't clear how this factored into his decision. He did have a specific QA plan which included the notion of early detection. His initial QA allocation was 20% of total staff.

40 days. The subject compared reported progress to elapsed time, and noted that he was already behind. He accurately assessed that causes could include hiring delays and an inexperienced workforce. He failed to notice the increasing requirements. He found that defect density exceeded the suggested range of 5 - 20 defects per kdsi. He accurately noted that this could be due to an inexperienced workforce, but he felt he was probably spending too much on QA, detracting from the coding effort and increasing cost.

80 days. The subject noted improvements in current staff and experience. He felt that he got a 'good response' to his change in QA allocation, noting density dropped to near the suggested range. He then noted his reported progress was still lagging elapsed time. In an effort to improve this he decided to further reduce his QA effort to 10%, but felt that he wouldn't be able to go much below that. In making this decision, the subject seemed to overlook the effect

that undiscovered defects would have on productivity and cost in terms of rework. Not having recognized the requirements increase, the subject elected to *decrease* staff by one, in an effort to improve productivity, with a smaller more experienced workforce. While the subject's decision demonstrated an accurate linkage between assimilation delays and productivity (Figure 4), he apparently misunderstood the long-term relationship between staff and cost.

120 days. The subject evaluated staff and workforce experience, and was pleased in the improved experience level. He observed a significant decline in defect density, which he accurately attributed both to the improved experience level and the decreased allocation to QA effort (Figure 5). He opted to increase effort back to the previous 15% level to avoid 'buggy code'. While this was probably an astute decision, he could have better ascertained the cause of the declining density by also examining the QA expenditure rate. If expenditures were declining as rapidly as density, it may have indicated that the 10% allocation was still acceptable.

160 days. During this period, the subject first recognized the growing requirements. Based on this, he immediately decided to increase the workforce from four to six. Based on defect density and expectations, he felt the QA allocation was acceptable, but cost and schedule definitely would increase. The new increased cost estimate was not mathematically supported, just an intuitive increase of 500 person-days based on requirements growth to date and expectations that requirements growth would continue. The new schedule estimate was obtained by dividing the days expended by the percent of code completed. This would have given the total duration required, but the subject mistakenly added it back to days expended to obtain an excessive estimate of 540 days. Fully accurate linkages between staff, cost and schedule were not evident at this point, however, the subject was beginning to demonstrate a partial understanding of the relationships.

200 days. The subject was satisfied with staff, experience, and defects. He also commented that his last schedule estimate was probably in error. Noting the additional requirements increase, his first inclination was not to increase staff because he felt the training time and additional overhead were too significant this late in the project. Reconsidering progress based on expectations that requirements would continue to grow at the same rate, he decided that he was in fact only a third of the way through the project. Given that deduction, he was more willing to change the workforce, and chose to increase staff from six to eight. He seemed to

become so engrossed in the staffing decision that he forgot the other two. Despite acknowledging the schedule estimate was probably in error, and expecting requirements to continue to grow, he changed neither schedule nor cost. QA also remained unchanged.

240 days. Again the subject was satisfied with the current staff, experience and defects. He re-estimated cost by dividing the person-days expended by the percent complete to get a new total cost. He also corrected his previous error in schedule estimation by computing resources required to complete the project and dividing by current staff. Ironically, the computation of resources required could have been used to more accurately update cost as well. The subject felt that changing staff again at this point would be counterproductive. He was also content to leave the QA allocation at 15%. His apparent understanding of the relationships between cost and schedule was more complete in this period. He also commented that any staffing increase at this point would probably increase the overall cost and duration of the project, demonstrating the staff/schedule relationship.

280 days. Staff and experience were favorably evaluated by the subject. He noted a decline in defects to something outside of his expected range, and used this as a basis for increasing QA allocation back to 20%. What he failed to recognize, was that the decline in density was most likely due to fewer defects, rather than a shortage of people assigned to QA. A comparison to QA expenditures would have helped make this apparent. The subject noted requirements growth had nearly ceased, and the percent reported complete increased acceptably. He re-estimated cost and schedule, and found neither needed to be updated.

320 days. This period, the subjects process was almost identical to the last. He did recognize that defect density continued to decline despite his additional allocation. He accurately deduced that his more experienced workforce was producing fewer defects, and he elected to reduce allocation back to 15%. With further analysis, he may have been able to lower it even further. Having recalculated his schedule estimate and finding it unchanged, the subject did not re-estimate cost.

Summary. While the subject started out with a QA plan, he managed the QA allocation level based on the defect density and expectations. By adjusting QA allocation and observing the effects, he was fairly effective in determining a workable level. Had he examined the trends in QA person-days per kdsi, he could have more effectively made decisions, and used

less of a trial and error approach. Despite an errant reduction in workforce early in the project and an excessive schedule estimate, the subject eventually demonstrated a fairly thorough grasp of the relationships between cost, schedule, and staff.

g. Subject 10 Narrative

Initial decision. The subject computed the underlying productivity assumption for the initial estimates and also asserted his expectations for individual productivity. The subject computed the staff size based on the initial cost and schedule estimates and also based on the initial requirements and schedule estimates, then used his own expectations for productivity. The subject opted for the more conservative estimates resulting from his productivity expectations. This was similar to the approach used by subject 7. The subject simply used the suggested QA level of 10% as his QA plan.

40 days. The requirements increase was recognized immediately, but the subject also recognized a productivity higher than his initial estimate. After considering the other factors mapped, he opted to absorb the requirements increase with the better than expected productivity, leaving staffing, cost, and schedule the same. This is consistent with the nominal map (Figure 6). Defects met the subjects expectations when compared to QA expenditures. Note the subject evaluated density against QA expenditures per kdsi, but did not use a predefined plan incorporating the notion that it is cheaper to detect and correct errors earlier.

80 days. Subject noted changes in requirements, productivity, experience, staff, and defect density. He also compared the percentage of the project completed to the percentage of resources expended. He then seemed to use this as a basis for increasing the duration by 25 days, however the complete source of this decision isn't clear. It seemed as though he simply sensed he was behind and applied an arbitrary 25 day extension to the schedule. Other decisions were unchanged.

120 days. Very similar to the previous period, except the comparison of resources expended and requirements completed appeared to be used as the basis for increasing the estimated project cost by 11 person-days. He noted that it was a choice between increasing cost or increasing staff, indicating a linkage between the two. At this point, the subjects

understanding of any other relationships between cost, schedule, and staff was not readily apparent.

160 days. This time the subject only noted the increase in requirements and the lag of tasks completed behind resources expended. He then increased staff by one.

200 days. During this period, the subject observed an increase in productivity, but did not trace it to perceived schedule pressure. He also noted increases in requirements and experience level, and assessed the defect density. He then reduced his QA effort to 5% of total staff. Although the complete basis for this decision was not clear, it seemed to be based on a perceived schedule pressure. This is represented by the 'Defect Expectations' on the map. The subject also made comments accurately linking an increase in staff to a decline in productivity, which would naturally increase cost. This is seen as the link from workforce experience to productivity in the nominal model (Figure 4). This appeared to be the basis for his unwillingness to add people late in the project. He also identified a tradeoff between staff and schedule which was partially accurate. He stated that a requirements increase could be addressed by either a staff increase or a schedule increase. He failed to note however, that an increase in staff could also lead to a schedule increase, especially late in the project.

Summary. At no time in the simulation was it clear that this subject understood the relationships between cost and schedule. In fact, he frequently used the concepts of days and person-days interchangeably. Many of his decisions seemed to be based on a gut feeling rather than a complete understanding of the relationships. QA decisions didn't seem to stem from a clear plan. Following the simulation, he commented that his overall QA strategy was to keep it within the range offered by the simulation as normal (5-20 defects per kdsi). At one point during the simulation, he compared QA expenditures to defect density. There was no evidence however that he understood the cost benefits of early detection, nor did he ever completely analyze the QA problem. Rather it seemed to be on the peripheral of his decision making.

F. SUMMARY OF OBSERVATIONS

By considering the survey results, decision and outcome data, and the individual decision processes, several key issues are evident.

1. Ability to cope with system complexities

Consistent with the literature, all of the subjects seemed to have difficulty in dealing with the complexities of the decision environment. They tended to deal only with small causal chains at a time, rarely connecting the chains to form the more complex causal webs which comprise the problem space. While some of the subjects mental models improved over the course of the project, true learning was frequently hampered by their misperceptions of the causal relationships. In several instances, subjects were observed to engage in justification of past decisions by forcing the current project status to fit into their causal model. This prevented them from seeing the actual relationships.

2. Quality assurance most difficult

While subjects had difficulty with all aspects of the project, the QA subsystem seemed to be the most perplexing. Subject 7 clearly voiced an understanding of the benefits of early detection, but clearly did not understand the costs of placing too great an emphasis on QA. Subject 9 had what was arguably the most effective strategy. He tried various QA levels and assessed the response of the defect density. Subject 1 felt that QA early was a waste of time. The remaining subjects placed little emphasis on QA or ignored it completely in their decision processes.

3. Experience effects

While seven subjects is an admittedly small sample, it appeared from this group that the more experienced managers made decisions differently than the others. Subjects 1 and 3 particularly were less able to voice their thoughts. This may be due to the fact that their decision processes have been so internalized that they are difficult to verbalize. These subjects also seemed to make decisions more on trends than specific data. They did fewer calculations to support their decisions, preferring to make adjustments which corresponded to the trends. They also seemed to be less likely to make changes at all, preferring to let things go to see what happens. Interestingly, the most experienced subject, subject 1, was the one who advocated sliding QA efforts until the end of the programming phase.

4. Goal priorities

The subjects were given the explicit goals of minimizing overruns in both cost and schedule. All of the data seems to indicate that the subjects placed a greater emphasis on schedule than cost. That is, they were more willing to go over budget than delay the delivery of the project. This was evident in the survey results, the decisions, and the outcomes.

5. Other observations

The survey results indicated subjects used the reports more than the graphs. The protocols indicate that those who used the graphs identified trends earlier. Subject 7 for example, never used the graphs at all. Had he looked at the plot of QA effort versus defect density he would have clearly seen that his effort was disproportionate to the defects detected. Others successfully used the status graph to identify when the growth rate had begun to decline. This potentially prevented them for adding further staff late in the project.

It was also observed that many of the subjects did not immediately identify the requirements growth. This clearly seemed to effect their willingness to increase staff. Had they all been aware of the change earlier in the project, their decisions may have been different.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Software projects in both the public and private sectors have been plagued with an inability to meet budgets and deadlines. Inadequate management tools and a general lack of understanding of the decision processes involved in managing software projects are largely responsible for many of the problems experienced.

The purpose of this research was to explore the decision processes of experienced software project managers. Towards that end an experiment was conducted which simulated the decision environment over the course of the project. The experiment included the capture of managers' decision and performance data, and their decision processes.

The results suggest that, consistent with the literature, managers have difficulty forming a complete mental model of the decision environment. The quality assurance subsystem posed an especially difficult problem.

While subjects all generally had difficulty with the environment, decision processes varied substantially between subjects. The experience level of subjects seemed to be a predictor of some of the variance. Subjects with the most experience apparently had difficulty verbalizing their decision processes, made fewer changes, and based their decisions on general trends vice specific calculations.

As would be expected, subjects refined their models of the decision environment as the management problem progressed. While learning from feedback was taking place, it was not clear that all learning was correct. In some instances, subjects seemed to adjust the feedback to fit their model of the environment rather than the other way around. This was frequently evidenced by the use of data to justify and rationalize past decisions.

Finally, the results indicated that subjects placed a higher emphasis on meeting schedule deadlines than budget constraints. This was indicated both by the survey results, and the analysis of the decision processes themselves.

B. RECOMMENDATIONS

Based on the analysis and observations made, several recommendations are appropriate:

- Conduct further research on the QA subsystem. The research can be tailored to developing strategies for dealing with the uncertainty of quality assurance, and assessing the effectiveness of training in quality assurance strategies.
- Conduct further research on the effects of varying degrees of experience on decision makers decision process. Particular emphasis should be placed on exploring how more experienced managers internalize their decision processes.
- Improve the visibility of the requirements changes in the simulation interface.
- Use simulation systems as training tools as well as research environments. Many of the subjects improved their understanding of the problem space in just this brief exposure. Comments indicated that they found the exercise useful and learned from it. Development of training environments could include more object-oriented metrics in the simulation interface to make it compatible with current development environments.

APPENDIX A

PROJECT QUESTIONNAIRE

Disk Number _____

1. In making your decisions, how much weight out of 100 points did you accord to the following goals? (The numbers should total 100 points.)

Cost _____

Schedule _____

100

2. How clear were the instructions regarding the task?

1	2	3	4	5	6	7	8	9
Not at all								Very
Clear								Clear

3. To what extent were the graphs of the progress of the project helpful in improving your decisions?

1	2	3	4	5	6	7	8	9
Not at all								Very
Helpful								Helpful

4. To what extent were the reports on the progress of the project helpful in improving your decisions?

1	2	3	4	5	6	7	8	9
Not at all							Very	
Helpful							Helpful	

5. Please give us some information about yourself (in absolute confidence. At no time will your name appear in the results. The data will only be used in an aggregate statistical sense).

a. Age (check one)

☐ under 30 ☐ 30-35 ☐ 36-40 ☐ 41-45 ☐ 46-50 ☐ 51-55 ☐ over 55

b. Work experience:

1.) Have you in the past participated in project management (Y/N)? ☐

2.) If YES, to what extent was the task in this simulation similar to your previous experience?

1	2	3	4	5	6	7	8	9
Not at all							Very	
Similar							Similar	

3.) How many years have you spent as a project manager? _____

4.) Please relate your project management experience to the best of your recollection.

- For project type, specify Embedded, Organic, Semi-detached or other.
- Embedded systems refer to those systems where the interfaces are rigidly defined, complex and inflexible (e.g., weapons systems).
- Organic systems are on the other end of the spectrum, having unrestrictive interfaces (e.g., administrative systems).
- Semi-detached are somewhere in the middle in terms of rigidity and complexity of interfaces.
- If other, explain.
- For language, list the language (FORTRAN, C, ADA, etc.).
- For platform, list the platform used, (e.g. super-computer, mainframe, workstation, PC, etc.).
- For project size, specify in delivered source instructions and/or staff size.

Project Type	Language	Platform	Project Duration (months)	Size (DSI or staff)	Remarks

5.) Please relate your experience as a developer, again to the best of your recollection.

Project Type	Language	Platform	Project Duration (months)	Size (DSI or staff)	Remarks

c. Education. Please provide your educational background.

Degree Level	Area of Study	Date
High School	NA	
Bachelors		
Masters		
Other		

d. How many hours (per week) do you use computers? _____

6. Your general comments regarding the simulation (use back if needed):

*** END OF SIMULATION ***

Thank you for your participation.

Questionnaire Results

Subj #	Cost/ Schedule Weights	Instruction Clarity	Graphs Helpful	Reports Helpful	Age	Task Similarity (0=no exp)	Year s Exp	Type of Software Project Management Experience	Education Summary	Computer Hours/ Week
1	40/60	9	3	9	over 55	4	18	Semi-detached; FORTRAN, BASIC, C/C++; Mainframe, PC, Workstation	BS(Edu)/ MS(Metro)	over 20
2	60/40	7	4	8	41-45	0	0	No management, development only	AA/(CS)	30
3	50/50	8	8	8	over 55	7	23	Semi-detached; FORTRAN; Mainframe, Super-Computer	BS/MS/ PrePhD	10
4	40/60	9	7	9	under 30	0	0	None	BS(PhySci)/ MS(Ocean)	40
5	50/50	9	1	9	41-45	8	7	Organic; C, FORTRAN, Assembly; PC, Workstation. Embedded; Assembly; Mainframe	BS(HR/CS)	20
6	40/60	9	6	8	over 55	3	5	Embedded; FORTRAN, Ada; Mainframe, Workstation	BS/MS(Phy)/ MS(Met)	40
7	20/80	8	5	8	41-45	6	12	Semi-detached; FORTRAN; Super-Computer	BS(Ocean)/ MS(Metro)	35
8	40/60	9	4	7	46-50	7	17	Semi-detached; FORTRAN, Assembly, C; Super-Computer, Mainframe, Workstation	BS(Eng)	25
9	30/70	9	7	7	46-50	7	3	Semi-detached; FORTRAN, Ada, C; Workstation	BS(Geol)/MS (CompSci)/ MS(Metro)	10-15
10	30/70	9	8	9	51-55	5	10	Embedded; FORTRAN; Mainframe, Workstation	BS(Geo)/ MBA	20

APPENDIX B

FINAL INSTRUCTIONS

Disk Number ____

A. INTRODUCTION

The exercise you are about to undertake is similar in many ways to flight simulators that pilots use to mimic flying an aircraft from takeoff at point A to landing at point B. Instead of flying an aircraft, though, the simulator mimics the programming phase of a real software project. In this simulation, you will be more than an observer. In fact, you will play the role of manager of the programming phase of the project. Specifically, your role will be to track the progress of the project by reviewing status reports and graphs available every two-month interval (40 working days) during the programming phase. As the manager, you must then make two staffing decisions:

1. The total number of staff you need. (You can hire additional staff, or decrease the staffing level as you deem necessary to complete your programming task successfully.)
2. Decide what percent of your total staff to allocate to the Quality Assurance activity to be conducted throughout the programming phase (e.g. to do inspections).

B. PROJECT

The project that you will manage happens to have been a real project conducted in a real organization. For the project, you will be given a project profile containing the following initial information:

- Estimated Size of the System: in Delivered Source Instructions (DSI)
- Estimated Cost of Programming Phase: in Number of Person Days
- Estimated Duration of Programming Phase: in Number of Work Days
- Size of initial Core Team: in People

The Core Team is a skeleton staff of software professionals who are there to ensure continuity between the requirements/design phase (which you may assume has just been completed), and the programming phase you are to manage.

The cost and schedule estimates are derived from a new off-the-shelf estimation tool, call it "NEW_TOOL", that has been recently acquired.

Historically, the defect density (i.e. number of defects detected during programming divided by the number of KDSI developed) has ranged from 5 - 20 Defects/KDSI.

C. YOUR TASK

Your task at every 40-day interval is to review the project's status, and make any necessary adjustments to the staffing level and its allocation. In order to do so, you may feel that is necessary to first adjust the project's cost and duration targets. The staffing decision should be done as follows:

1. Decide on the total staffing level, and
2. Decide on what percentage of the staff should be allocated to the quality assurance function (i.e. a number between 0 and 100).

D. YOUR GOAL FOR THE TASK IS TWOFOLD:

Minimize overruns in both cost and schedule.

E. SOME IMPORTANT POINTS TO CONSIDER IN MANAGING YOUR TASK

1. As the manager of the programming phase, you specify the desired staffing level. You may find that your actual staffing level (as it will appear in the reports) is different from what you requested. This would be due to factors you cannot control, such as hiring delays and turnover.
2. The staff size that you select, and which appears in reports, may show fractions (e.g. 4.5 people) since people are allowed to work on more than one project.
3. When requesting additional staff, expect a delay in hiring. For modest additions to your staffing, the average hiring delay will be around 40 days. However, if you request a large number of additional staff, the average hiring delay will be much longer.
4. Once new people are hired, they must be trained and assimilated. The assimilation/training period is typically 80 days. During this assimilation/training period you can expect the new employee to be only half as productive as an experienced employee.

5. Adding more people increases communication and coordination overhead as happens in reality.

F. RULES OF THE GAME

1. If you have a question, ask the lab attendant.
2. You are not allowed to bring any notes to use during the simulation. Feel free to write on the documentation sheets provided.
3. A calculator is allowed and recommended.

G. INSTRUCTIONS FOR STARTING THE SYSTEM

Follow the instructions Carefully. If any problems arise, **immediately** seek out the lab attendant.

1. Insert the disk into the B: drive. Do not remove the disk from the drive.
2. From the C:\ prompt, type B:
3. Start the simulation by typing START at the B:\ prompt.
4. Follow the instructions as they appear on the screen.
5. The simulation is complete when the **% Programming Reported Complete** in the PROJECT STATUS REPORT is 100%. When this occurs **Call the lab attendant.**

DOCUMENTATION SHEET

Disk Number ____

REMEMBER YOUR GOALS ARE:

Minimize overruns in both cost and schedule.

INITIAL ESTIMATES:

Project Size	15,860 DSI
Project Cost	944 Person Days
Project Duration (start-end)	272 Days

TIME ELAPSED (DAYS)	ESTIMATED COST (PERS-DAYS)	ESTIMATED DURATION (DAYS)	STAFFING LEVEL (PERSONS)	QUALITY ASSURANCE (PERCENT)
Initial Decision	944	272		
Time Elapsed - 40 Days				
Time Elapsed - 80 Days				
Time Elapsed - 120 Days				
Time Elapsed - 160 Days				
Time Elapsed - 200 Days				
Time Elapsed - 240 Days				
Time Elapsed - 280 Days				
Time Elapsed - 320 Days				
Time Elapsed - 360 Days				
Time Elapsed - 400 Days				
Time Elapsed - 440 Days				
Time Elapsed - 480 Days				
Time Elapsed - 520 Days				

**** WHEN YOU ARE DONE, CALL THE LAB ATTENDANT ***

APPENDIX C

A. THE CODING SCHEME

1. The Trial Experiment

As mentioned previously, a trial experiment was conducted prior to running the experiment on the 10 expert subjects. The purpose of the trial experiment was twofold. First, to gain experience and evaluate the mechanics of the experiment. For this reason, the same procedures were followed in the trial experiment as the final experiment. Second, and more importantly, to obtain a protocol for use in the development of a coding scheme.

The subject for the trial experiment was selected based upon both his education and experience. As a graduate student in Information Technologies Management, the subject had received formal education in the field of software engineering. Combined with his experience as a professional software developer, this education provided an appropriate background in the domain area, and eminently qualified him as a subject for this experiment.

2. Developing the Code

Coding categories were developed by analyzing the single subject protocol from the trial experiment. To allow greater flexibility in the coding of the protocols, the coding scheme was segmented into categories, actions on categories, and up to two modifiers of each category/action combination. Video was used to clarify a subject's intent when statements alone were insufficient, or to otherwise establish the context of a statement.

There were a total of seven categories used: Information, Goal, Decision, Link, Task, Execute, and Remark. The bulk of the substance is contained in the first four categories, with the last three providing administrative detail. Each category and its associated actions, modifiers, and abbreviations are described below, followed by a summary and a description of modifier types. In some cases, examples are given to provide clarification. Reference to Appendix ??? will provide further illustration of the use of the various codes.

a. Information (I)

The Information or 'I' category was used on all statements which dealt directly with information or data points, excluding the specific goal or decision data points. Actions on

information, their abbreviations, modifiers, and descriptions are listed in the format:

Abbreviation - Action (Modifier 1, Modifier 2): Description.

- A - Acquire (Data Point, Source): Details specific information acquired and the location of the data (report or graph).
- M - Manipulate (Data Point, Operation): Indicates the data resulting from some operation.
- E - Evaluate (Data Point, Result): Indicates some analysis of a data point, and the general result (hi, low, or acceptable).
- R - Recall (Data Point, Period): Indicates recall of a previously known data point, and the period from which it was recalled (current period, previous period, first period, etc.).

The following excerpt from the trial subject illustrates the acquisition, manipulation, and evaluation of information:

STATEMENT	CAT	ACT	MOD1	MOD2
We've got - project size has gone up again to 24346.	I	A	SIZE	REP
	I	E	SIZE	UP
So at 24346 minus 18635	I	A	DSI	REP
we've got 5700 source instructions to go	I	M	RSI	SUB

The first statement indicates both the acquisition and evaluation of the project's size (MOD1 = SIZE). It was acquired from a report (MOD2 = REP), and the result of the evaluation was that it had gone up (MOD2 = UP). The next statement indicates the acquisition of the delivered source instructions (MOD1 = DSI) from a report, followed by a manipulation of previously acquired or recalled variables to produce remaining source instructions (MOD1 = RSI) using the subtraction operator (MOD2 = SUB). An information recall would be similar to acquisition, but used on information that had already been acquired, and was simply being recalled.

c. Goal (G)

The Goal or 'G' category was used to code all statements dealing directly with one of the two goals, schedule and cost. Actions on goals, their abbreviations, modifiers, and descriptions are listed in the format: *Abbreviation - Action (Modifier 1, Modifier 2): Description.*

- U - Update (Goal, Change): Indicates a conscious adjustment or reaffirmation of one of the two goals, and the direction of change (up, down, or no change) since the last time the goal was visited, either the last update or the last finalization.
- R - Recall (Goal, Period): Indicates a recall of a previous goal and the period of that goal.
- J - Justify (Goal): Indicates an attempt to rationalize or justify an action on a goal.
- E - Evaluate (Goal): Indicates some evaluation of a goal against actual performance.
- F - Finalize (Goal, Change): Indicates finalization of a goal for that period, and the change from the last periods final goal.

The following example illustrates a goal sequence from the trial subject's protocol.

STATEMENT	CAT	ACT	MOD1	MOD2
Simply for simplicity for this round, I'm going to go ahead and leave the estimated duration constant and not alter it as well.	G	U	SKED	NC
I don't want to alter too many variables at one time or I won't be able to tell what is effecting what.	G	J	SKED	
I'm going to go ahead and bump my staff up because the size of the project has already increased. So 6 people	D	I	RSTAF	UP
times 272 days	G	R	SKED	t-1
gives us a budget of 1632 person days	G	U	COST	UP
which is already an overrun of roughly 73% increase in cost.	G	E	COST	

The sequence indicates an update to the schedule goal in which the subject chooses not to change the schedule. This is followed by a justification for the choice. The subject then makes a staffing decision, followed by a recall of the schedule goal from the prior period (t-1). This information is then used to update the cost goal and evaluate that update. Shortly after this sequence, the goals for the period were finalized (see Appendix A).

d. Decision (D)

The Decision or 'D' category was used on all statements which dealt directly with either of the two decisions, total staff or percent of staff allocated to QA. Actions on decisions, their abbreviations and modifiers are listed in the format: *Abbreviation - Action (Modifier 1, Modifier 2): Description.*

- I - Initialize (Decision, Change): Indicates the initial decision of a period and the change from the previous decision.
- A - Adjust (Decision, Change): Indicates a conscious adjustment or reaffirmation of one of the two decisions, and the direction of change (up, down, or no change).
- R - Recall (Decision, Period): Indicates a recall of a previous decision and the period of that decision.
- J - Justify (Decision): Indicates an attempt to rationalize or justify an action on a decision.
- F - Finalize (Decision, Change): Indicates finalization of a decision for that period, and the change from the last periods final decision.

Examples of decision statements are similar to those of goals with one exception. The 'update' action from the 'goal' category was essentially broken down into initialize, and adjust. This was done to better account for the more structured decision sequences. Appendix A provides numerous specific examples of the coding of decision statements.

e. Link (L)

The Link or 'L' category was used to denote statements where the subject identified a causal link between two data points. Two distinctions were made in the actions, shallow vs. deep, and accurate vs. inaccurate. Patterned after Schweiger (1985), these distinctions allow closer scrutiny of subjects' causal analysis.

'Shallow' indicates an analysis only one level deep. For example, the statement, "but the workforce is 81% experienced - that contributes to the productivity being so high.", indicates identification of a shallow link between workforce experience and productivity. A deep link could be multiple levels deep or linking two causes to one effect. Using the previous example, if the subject had also tied in staff size (and the corresponding communication overhead) to productivity, the analysis would be classified as deep. If deep causal analysis is identified, additional data points can be included in additional modifiers as necessary.

The 'accuracy' of the subjects analysis is also captured in the coding scheme. A code of 'accurate' indicates a match between the causal relationship the subject identified and the software dynamics model, while 'inaccurate' indicates the causal link identified by the subject did

not match the model. It is important to note that the 'accuracy' is not a judgment of correctness, but rather a comparison to an existing model of the environment.

In some cases, the subject only indicated an awareness of a causal relationship, but not an actual identification of the linkage. This awareness is also captured in the coding scheme as described below. Actions on goals, their abbreviations, modifiers, and descriptions are listed in the format: *Abbreviation - Action (Modifier 1, Modifier 2): Description*. The two modifiers were used to identify the data points or variables involved; the first being the 'cause, and the second being the 'effect'.

- IDSA - Identify Shallow Accurate (Data Point, Data Point): Used when a subject identified a shallow causal relationship between two variables, and the relationship matched the model.
- IDSI - Identify Shallow Inaccurate (Data Point, Data Point): Used when the subject identified a shallow relationship which did not match the model.
- IDDA - Identify Deep Accurate (Data Point, Data Point, etc.): Used when a subject identified a deep causal relationship between multiple variables, and the relationship matched the model.
- IDDI - Identify Deep Inaccurate (Data Point, Data Point, etc.): Used when a subject identified a deep relationship between multiple variables, but the relationship did not match the model.
- ID - Identify (Data Points optional): Indicates subject was aware of a causal relationship, but was not sure or not clear about the details of the linkage.

f. Task (T)

The Task or 'T' category was used with only one action, that is, Select (S). This category action had no modifiers and was used simply to denote statements in which the subject chose the next task to act on. For example, "See how staffing's doing.", is merely a comment by the subject to indicate what they will do next.

g. Execute (E)

The Execute or 'E' category simply provided a break between periods to facilitate single period data analysis. No actions or modifiers were used. In cases where the subject didn't specifically state they were executing, a blank statement was added with the 'E' code to enable data queries.

h. Remark (REM)

The Remark or 'REM' category was used to code statements that had no relevance or were redundant. No actions or modifiers were used.

i. Coding Summary

Table 2 provides a summary of the coding categories, actions and modifiers used in coding the protocols.

CATEGORY	ACTION	MODIFIER 1 TYPE	MODIFIER 2 TYPE
Information (I)	Acquisition (A)	Data Point	Source
	Manipulation (M)	Data Point	Operation
	Evaluate (E)	Data Point	Result
	Recall (R)	Data Point	Period
Goal (G)	Update (U)	Goal	Change
	Recall (R)	Goal	Time
	Justify (J)	Goal	
	Evaluate (E)	Goal	
	Finalize (F)	Goal	Change
Decision (D)	Initialize (I)	Decision	Change
	Adjust (A)	Decision	Change
	Recall (R)	Decision	Time
	Justify (J)	Decision	
	Finalize (F)	Decision	Change
Link (L)	Identify Shallow Accurate (IDSA)	Data Point	Data Point
	Identify Shallow Inaccurate (IDSI)	Data Point	Data Point
	Identify Deep Accurate (IDDA)	Data Point	Data Point
	Identify Deep Inaccurate (IDDI)	Data Point	Data Point
	Identify (ID)	Optional	Data Points
Task (T)	Select (S)		
Execute (E)			
Remark (REM)			

Table 2. Coding Summary

j. Modifiers

Table 3 contains an exhaustive list of all the possible values for each of the modifiers used and an associated description.

MODIFIER	VALUES	DESCRIPTION
DATA POINT	STAFR	Required staff
	CSTAF	Current staff
	WFE	Work force experienced
	TPRODR	Total productivity required
	TPRODC	Total productivity current
	IPRODR	Individual productivity required
	IPRODC	Individual productivity current
	SIZE	Total requirements (delivered source instructions)
	DD	Defect density
	NDD	Normal defect density
	PDEXP	Person days expended
	%COMP	Percent of requirements completed
	INCSZ	Percent increase in project size (requirements)
	HDEL	Hiring delay
	DR	Days remaining based on elapsed time and estimated schedule
	PSI	Projected source instructions with current staffing, productivity, and days remaining
	DSI	Delivered source instructions
	RSI	Remaining source instructions left to deliver
	TMELAP	Time elapsed
SOURCE	REP	Report
	GRAPH	Graph
OPERATION	ADD	Addition
	SUB	Subtract
	DIV	Divide
	MULT	Multiply

Table 3. Modifiers

MODIFIER	VALUES	DESCRIPTION
RESULT	HI	Higher than expected
	LO	Lower than expected
	OK	As expected or acceptable
	CONV	Change the units or form of the data
PERIOD	t	Current period
	t-1	Previous period
	t ₀	First period
	EXP	Experience
	GEN	General case (used with GOAL RECALL)
GOAL	COST	Pertaining to the cost goal
	SKED	Pertaining to the schedule goal
DECISION	RSTAF	Requested staff
	QA	Percent of staff allocated to QA

Table 3 (continued). Modifiers

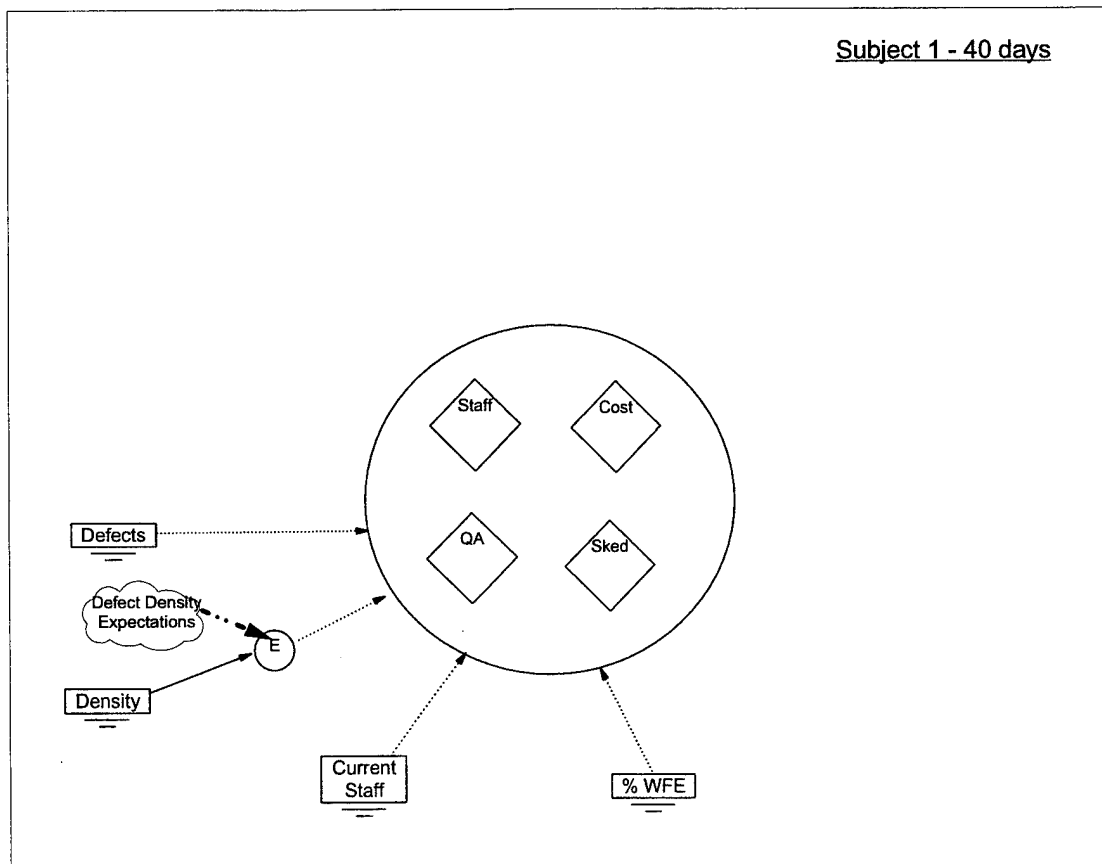
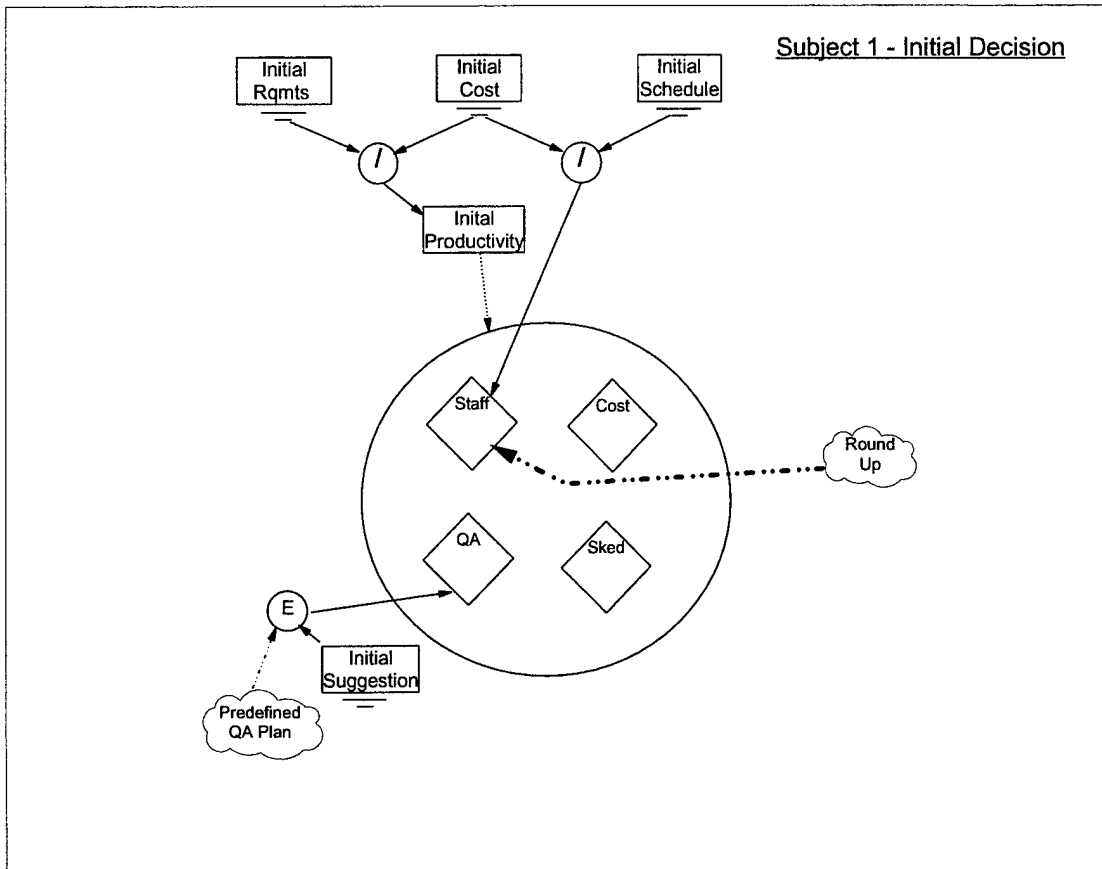
3. Coding the Protocol

To further illustrate the coding process, the single-subject protocol and the codes for each statement are included in Table 4. For reasons of anonymity and space, this is the only protocol included in its entirety.

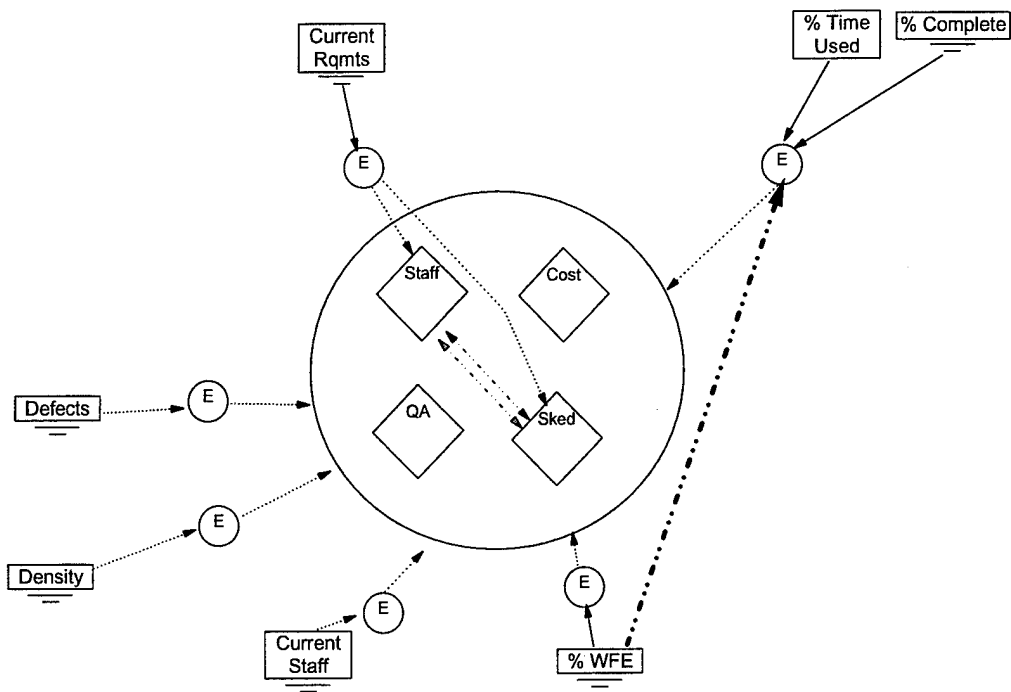
4. Using the Coded Protocol

From the coded protocol, a number of analysis can be conducted. Many of these were introduced in Chapter II, including frequency analysis and transition analysis. Frequency analysis entails determining the frequency each task can be performed. Transition analysis is concerned with the sequence tasks are performed in. This can be expended to include an analysis of which tasks preceded and followed each task category. By adding a time element to the code, a task duration analysis can be conducted. By using the above coding scheme in a spreadsheet, these analyses can be semi-automated. The hierarchical nature of the scheme lends itself to analyses at varying degrees of granularity.

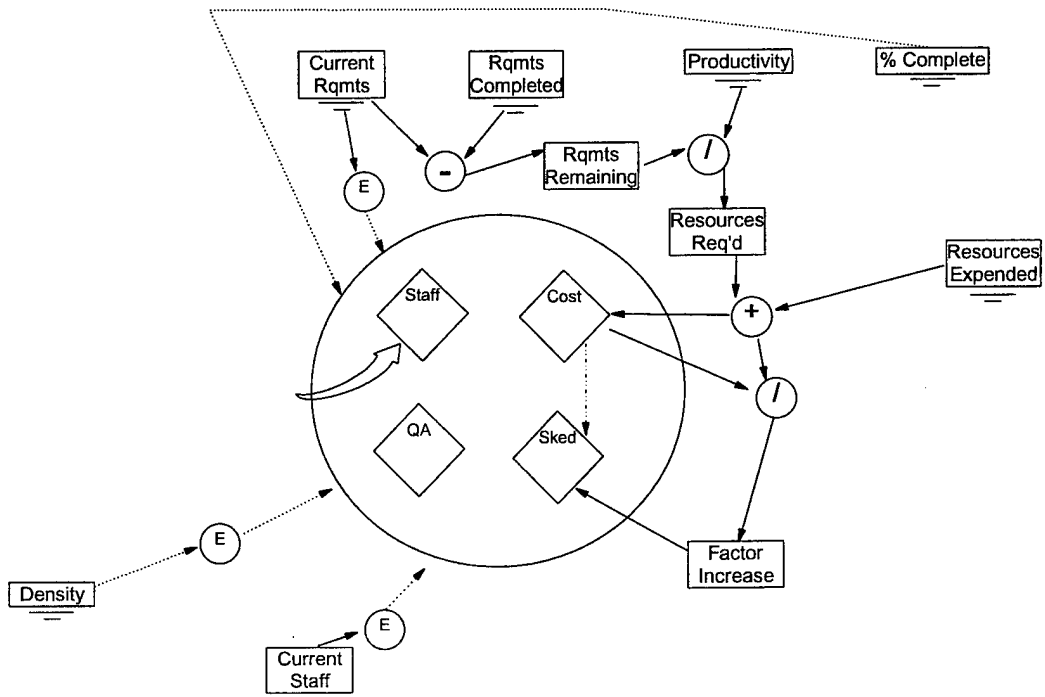
APPENDIX D

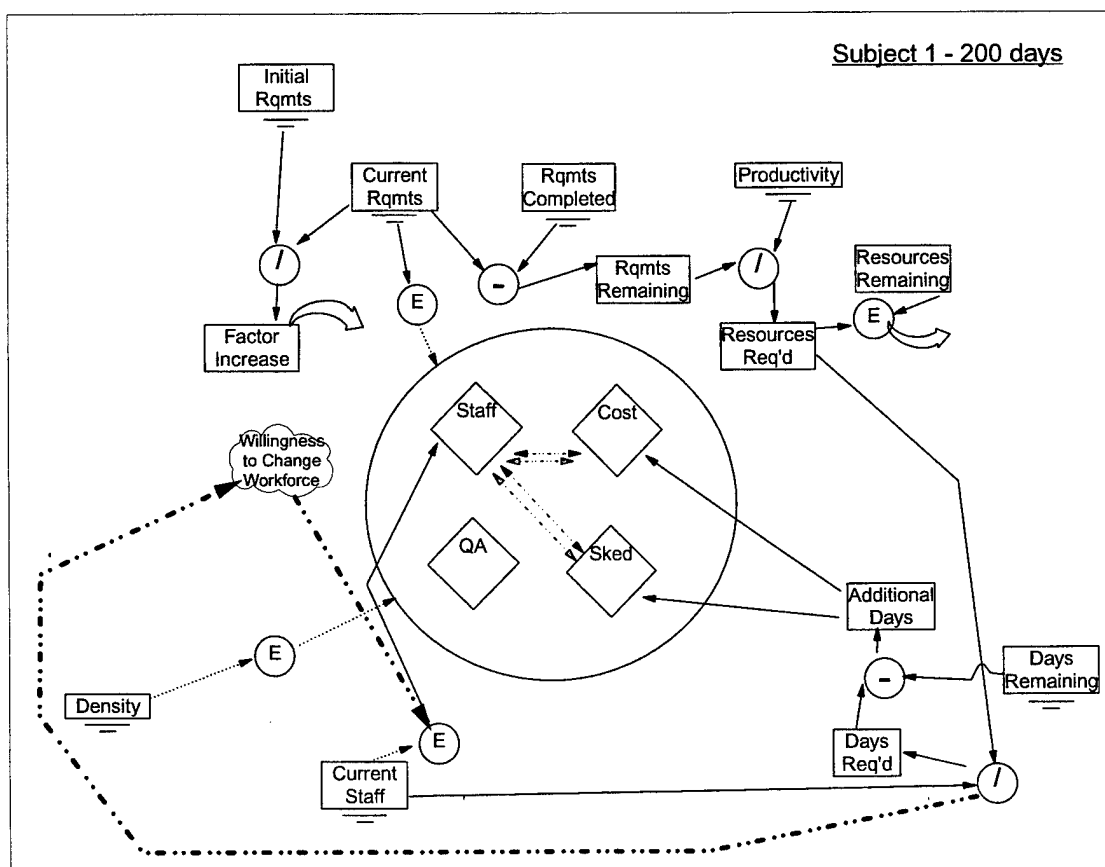
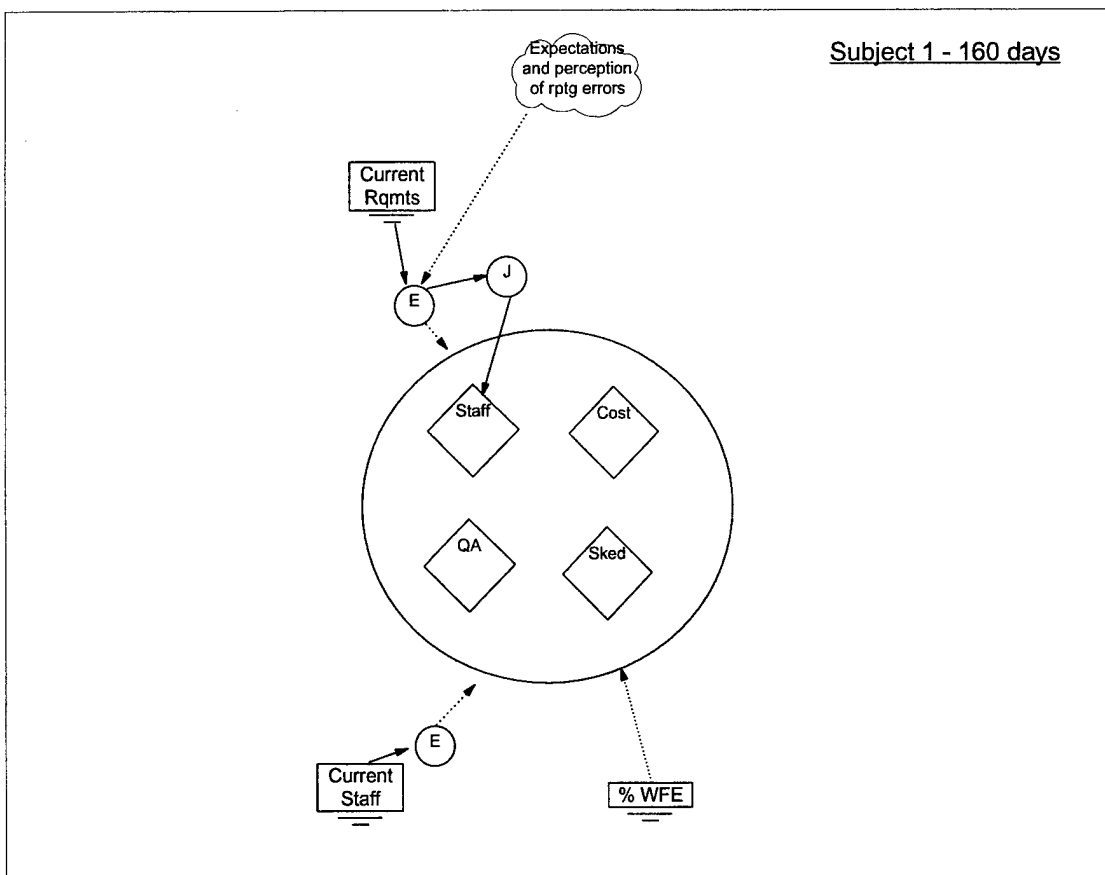


Subject 1 - 80 days

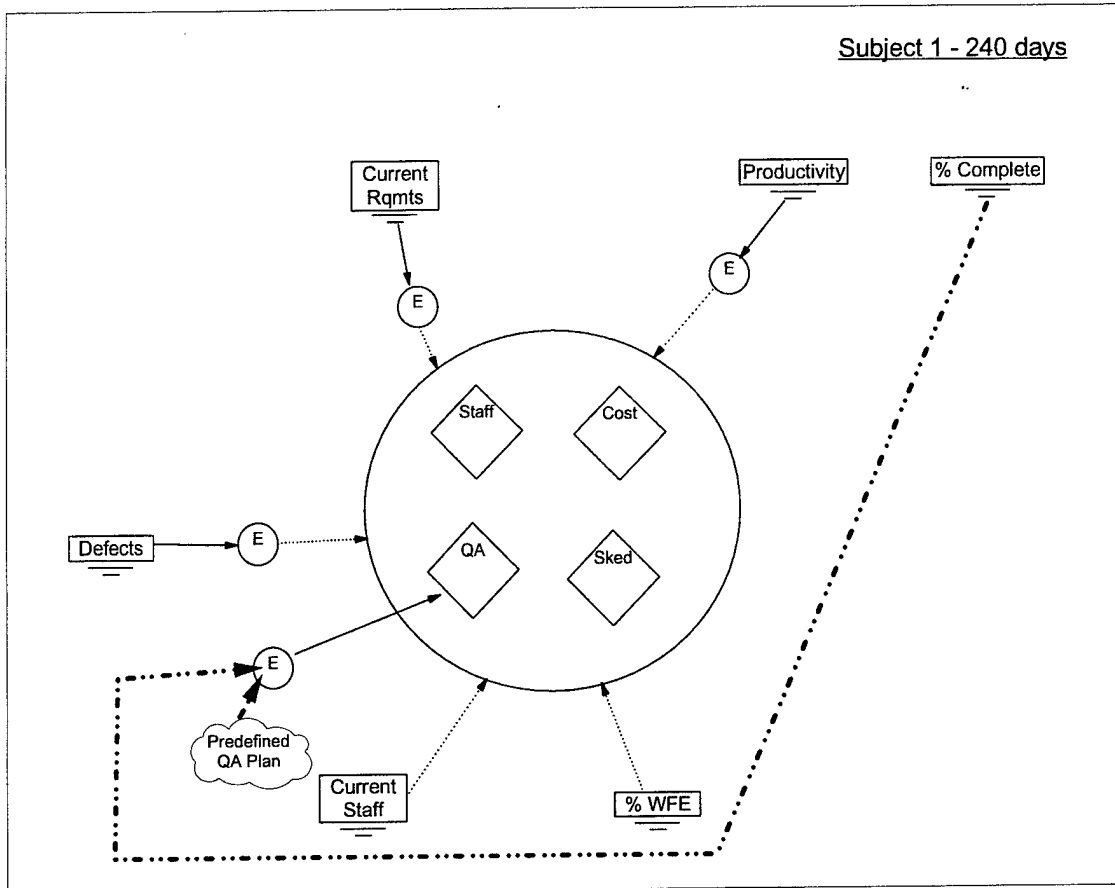


Subject 1 - 120 days

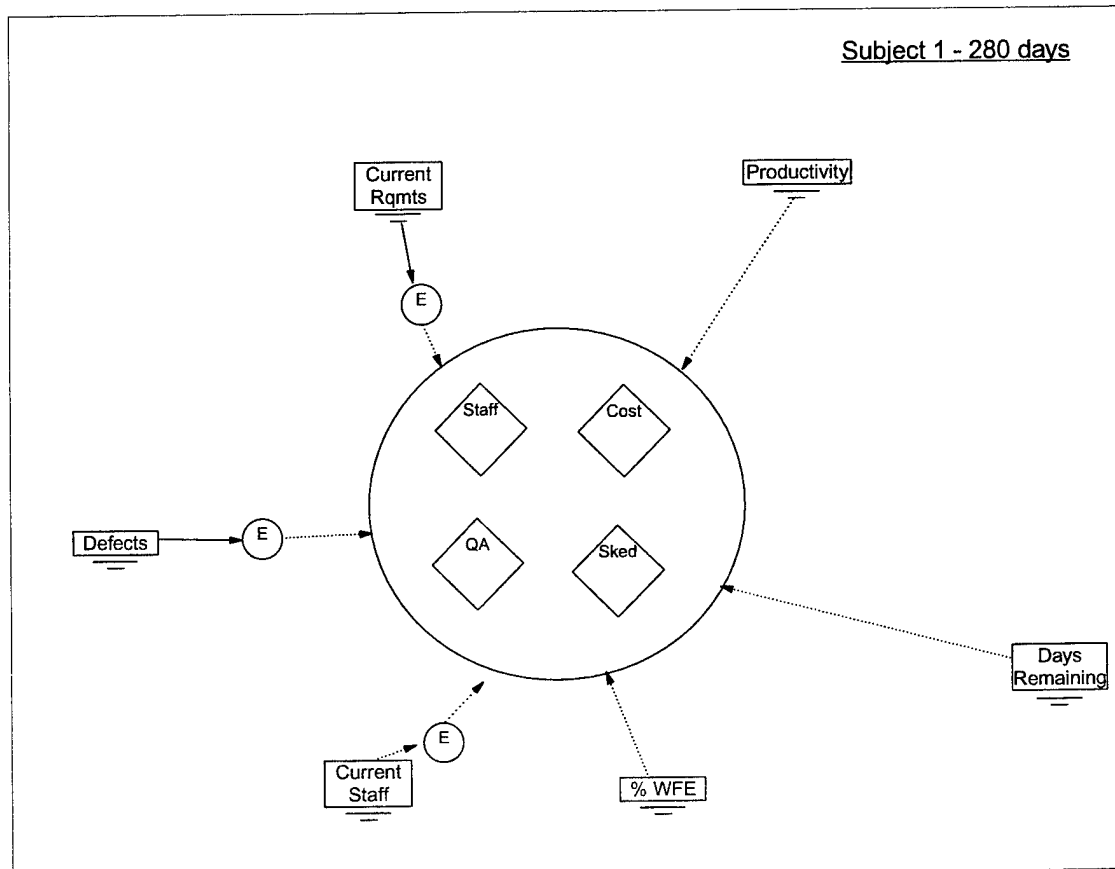




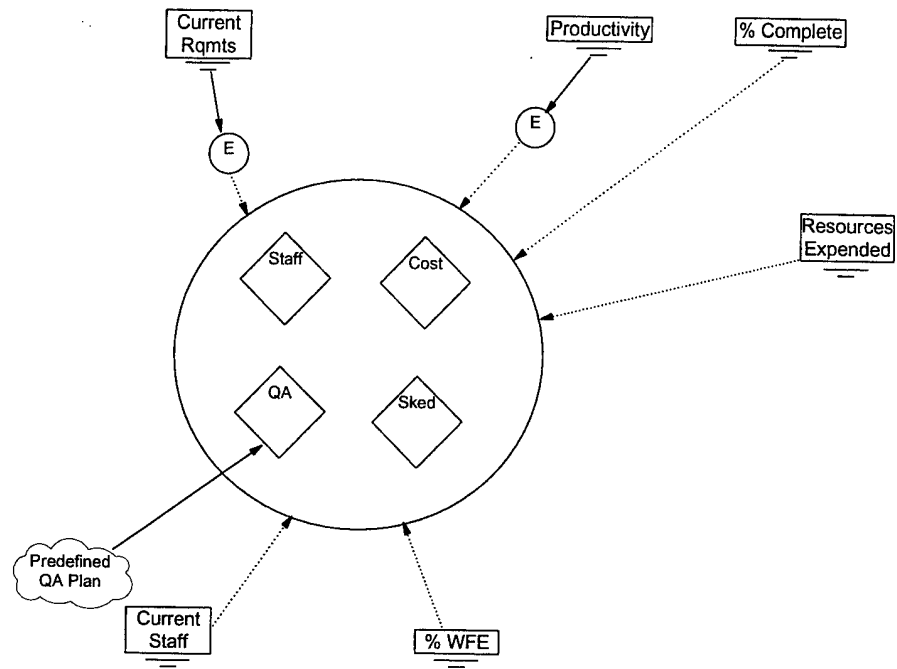
Subject 1 - 240 days

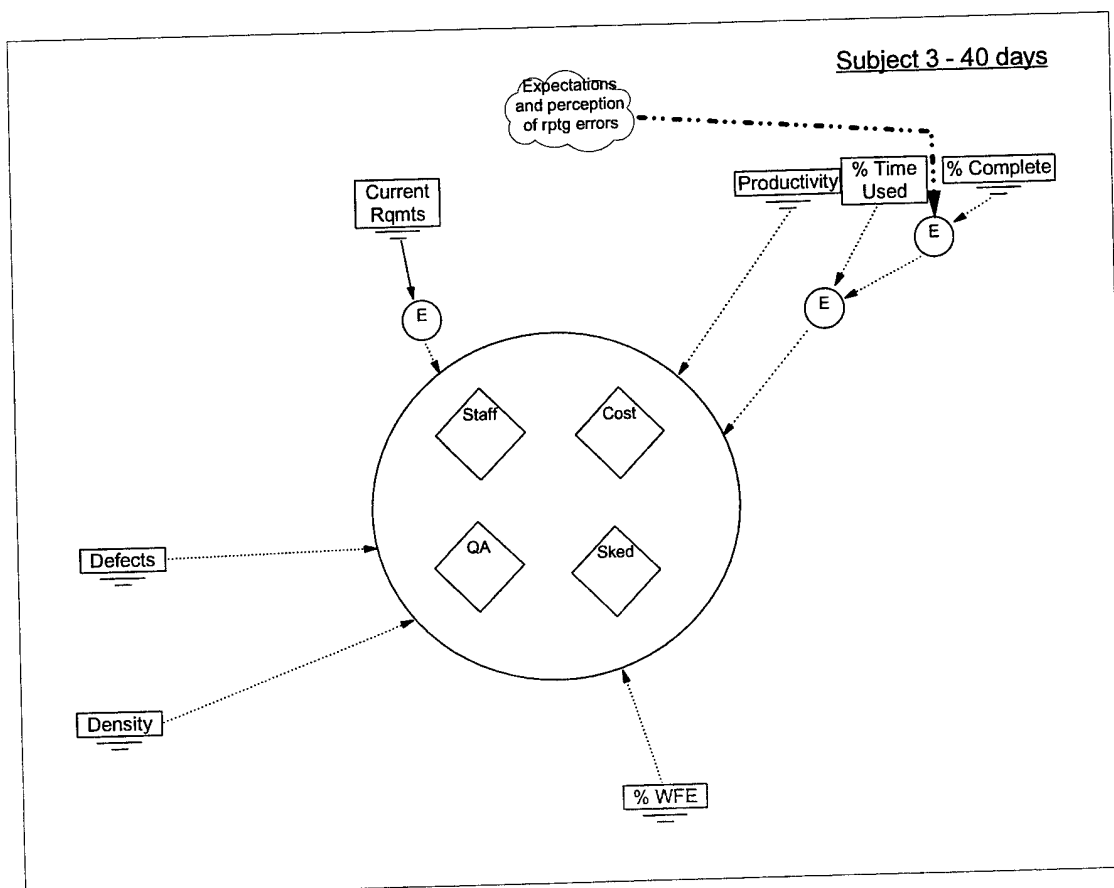
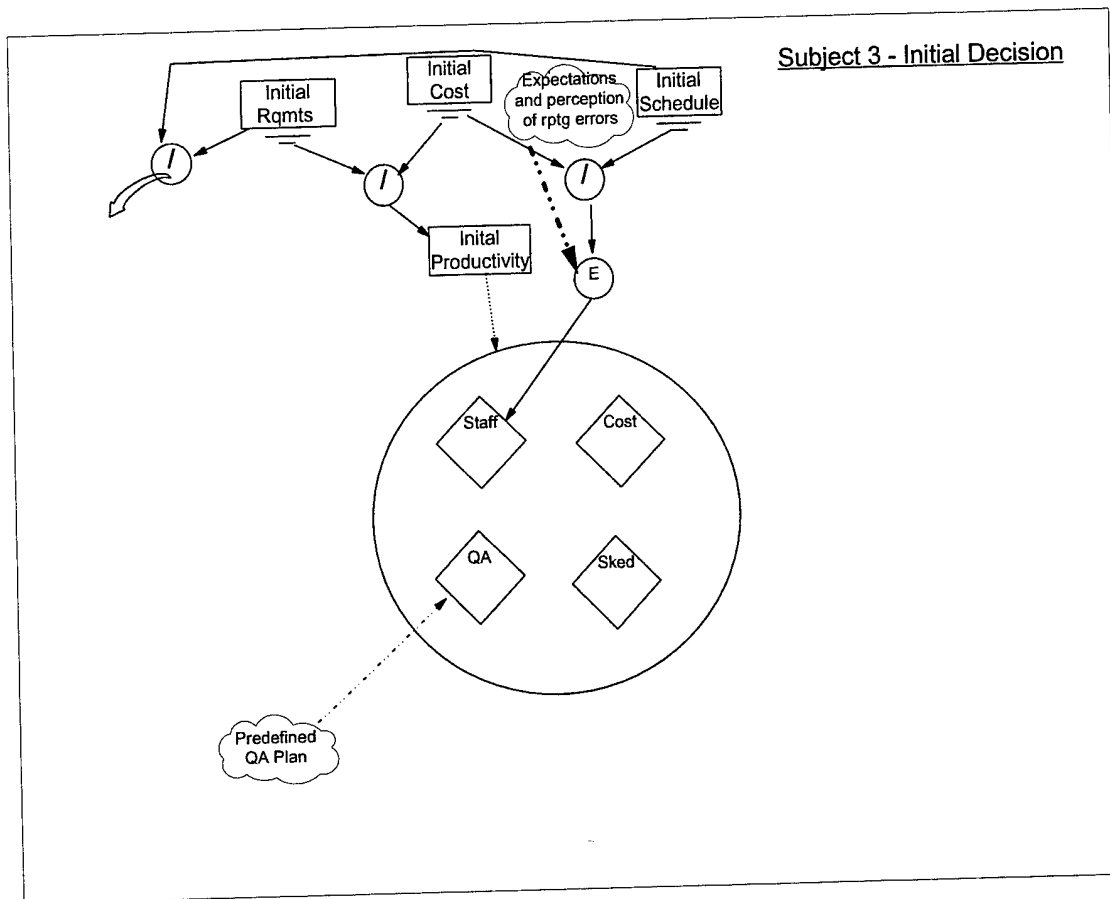


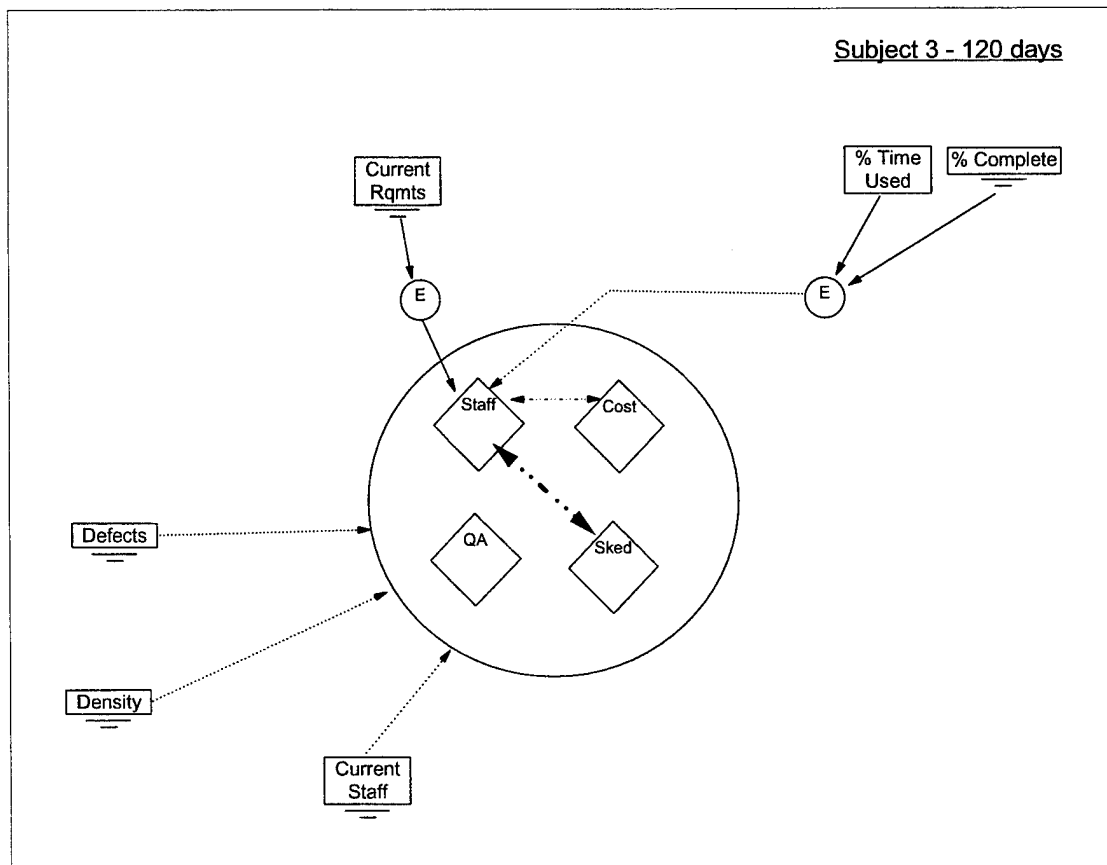
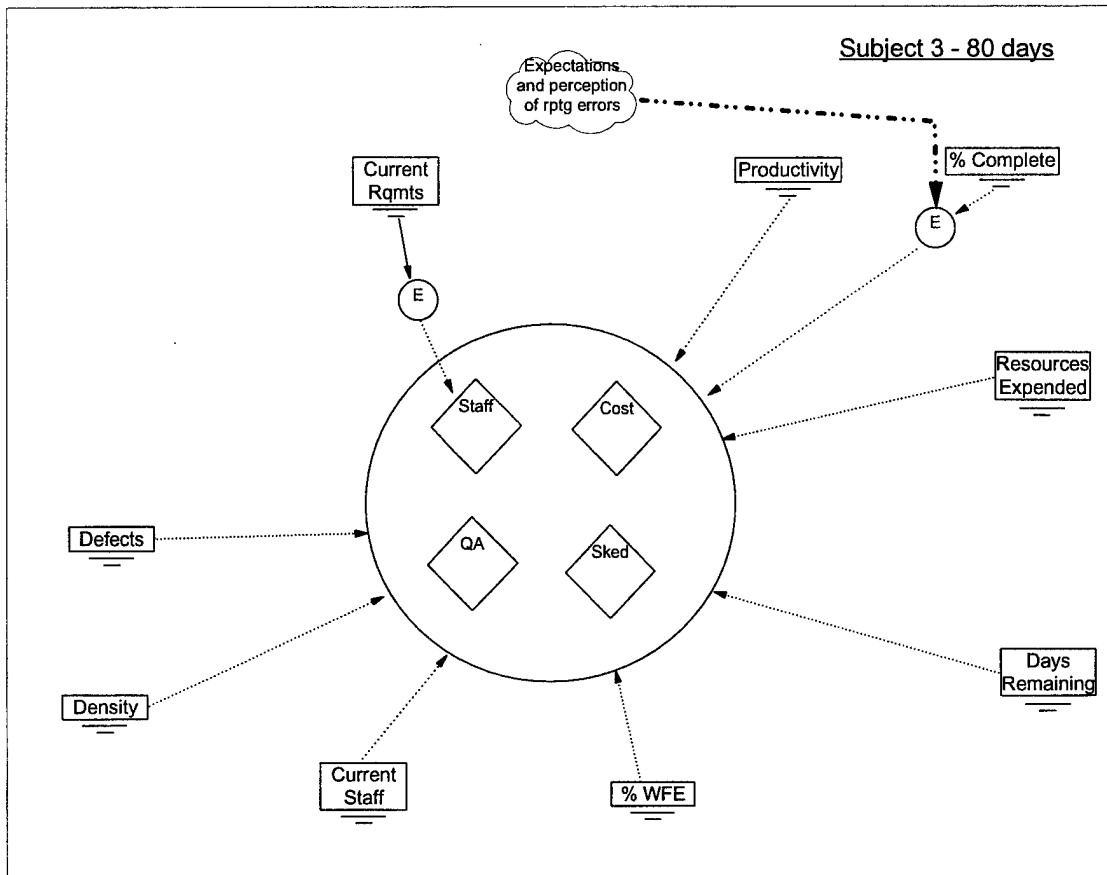
Subject 1 - 280 days

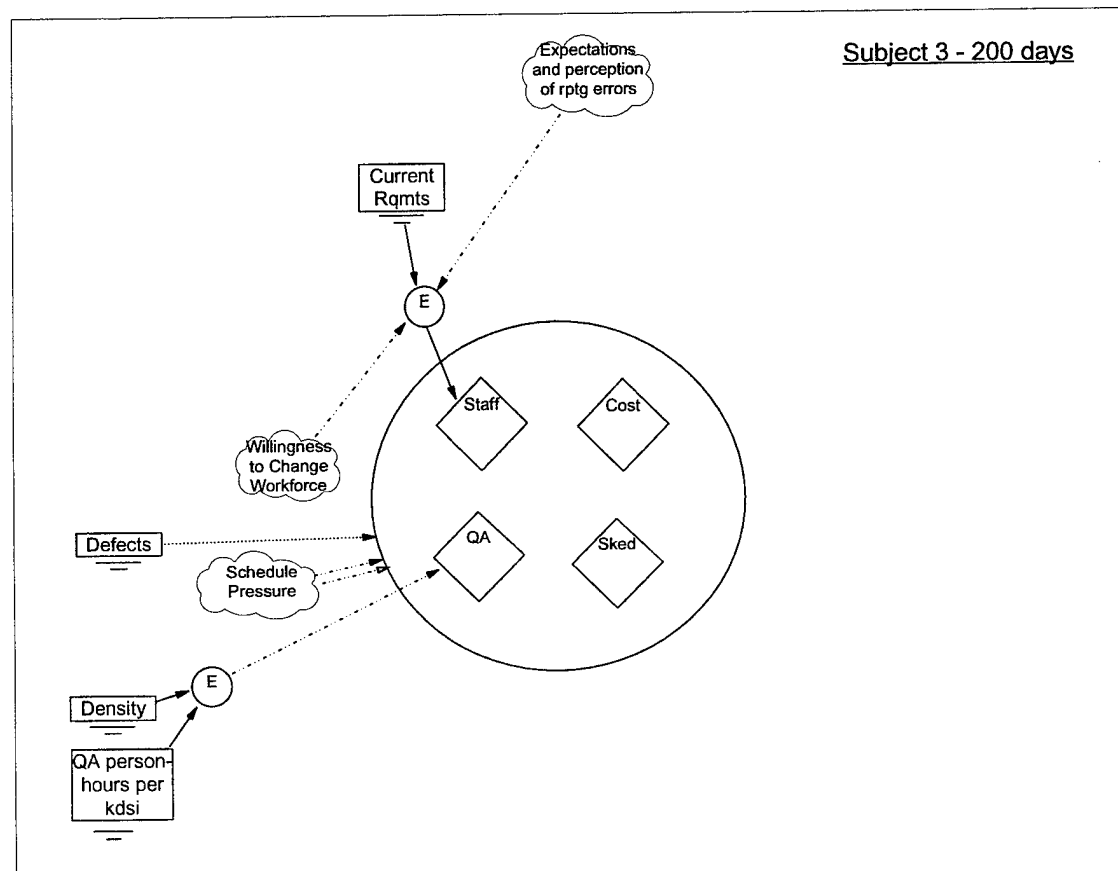
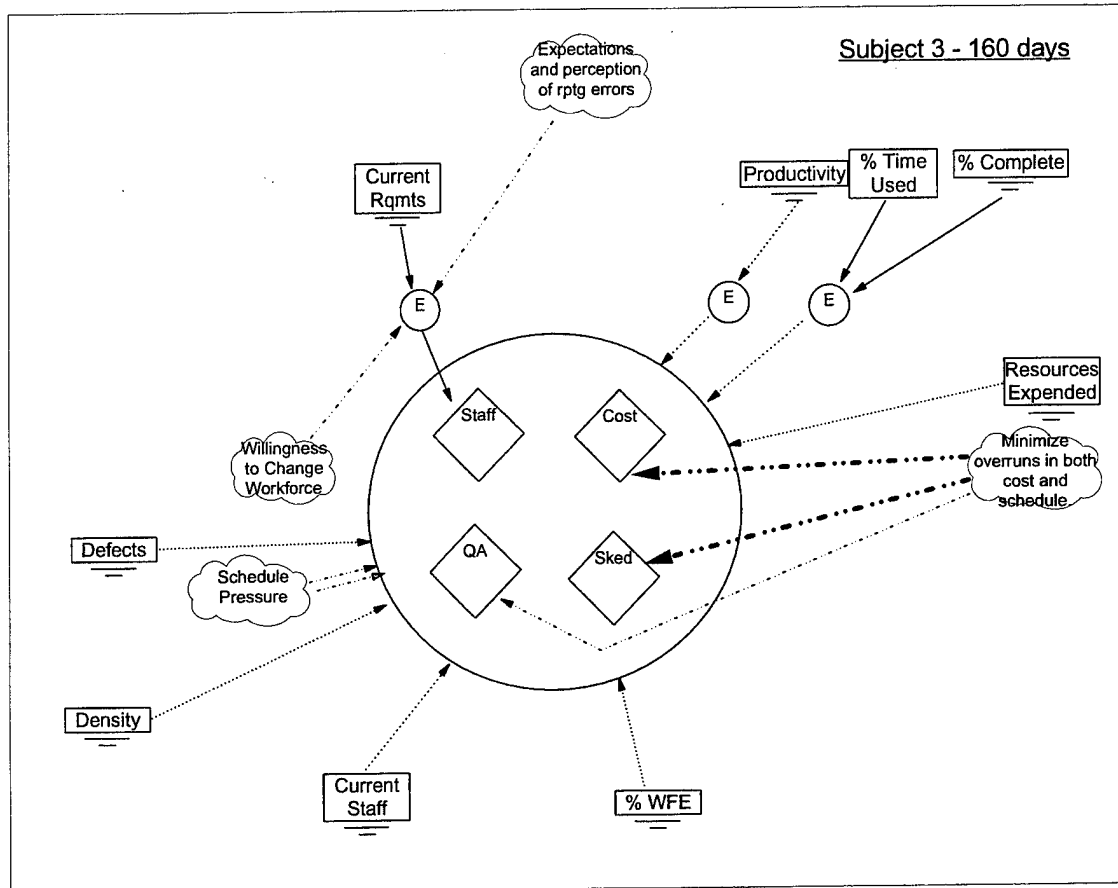


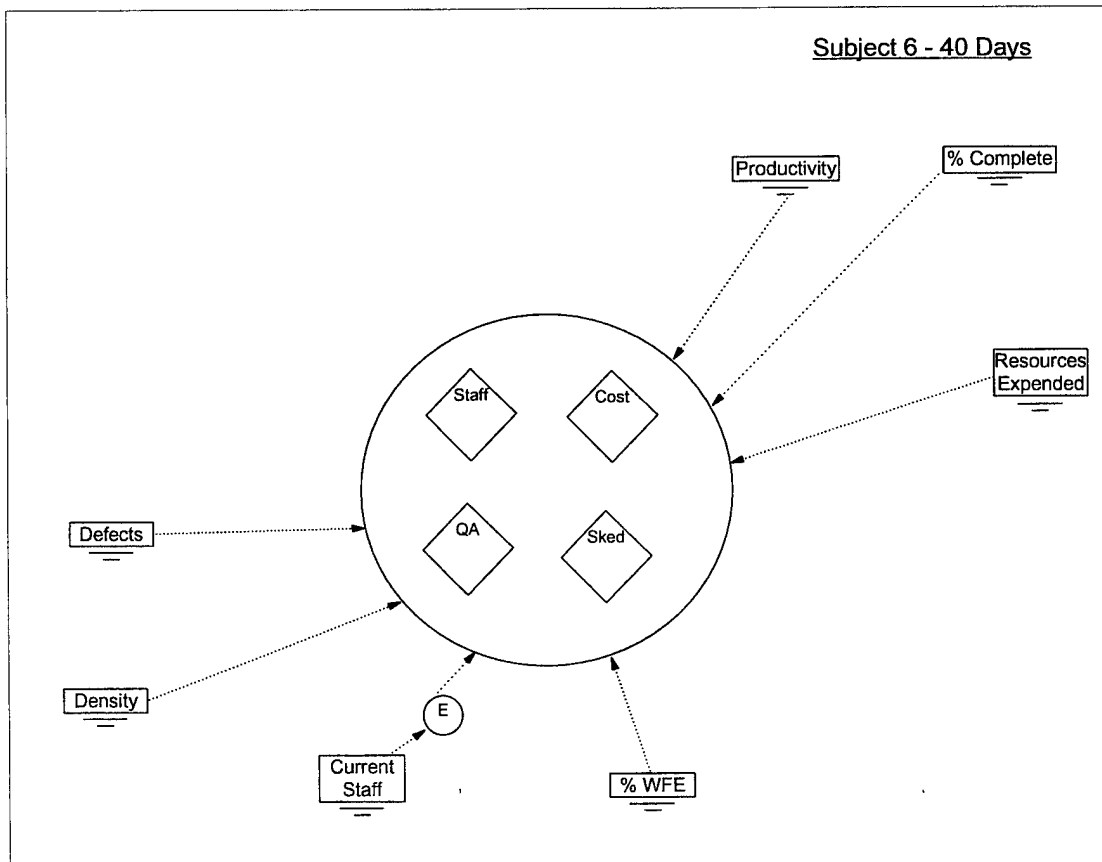
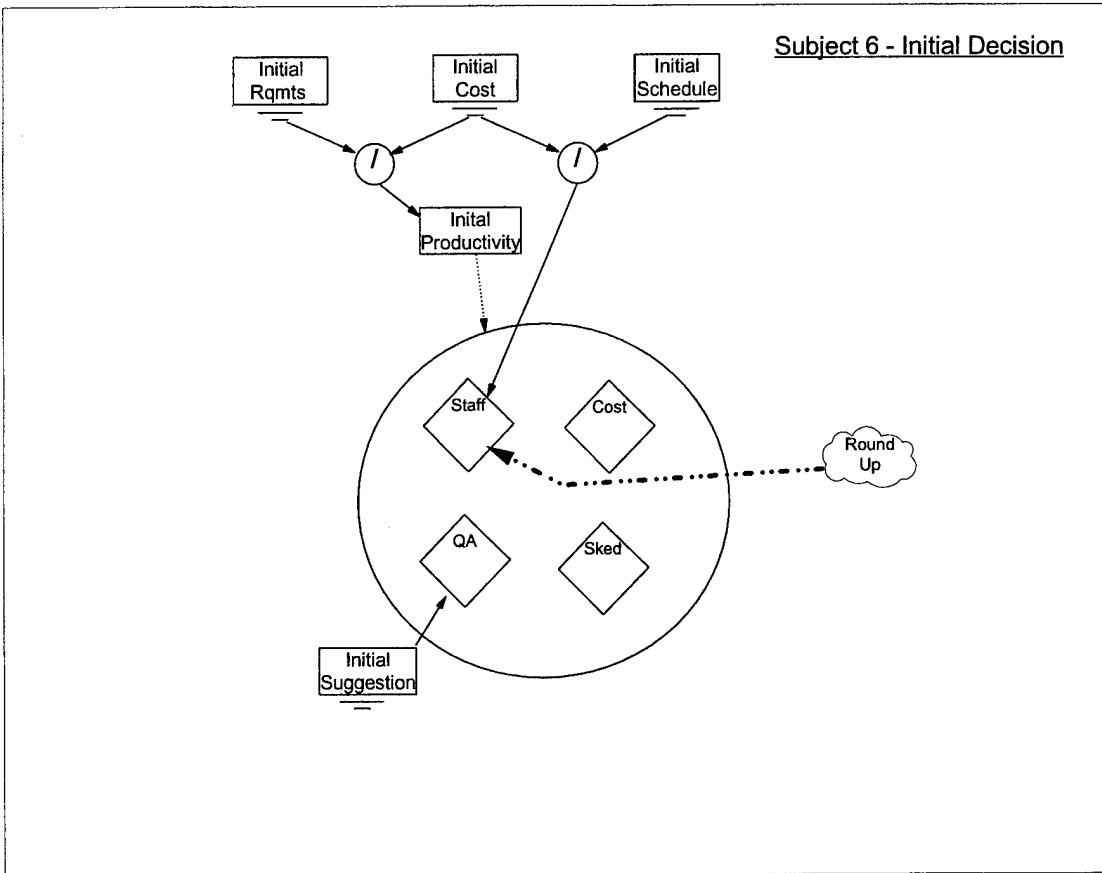
Subject 1 - 320 days

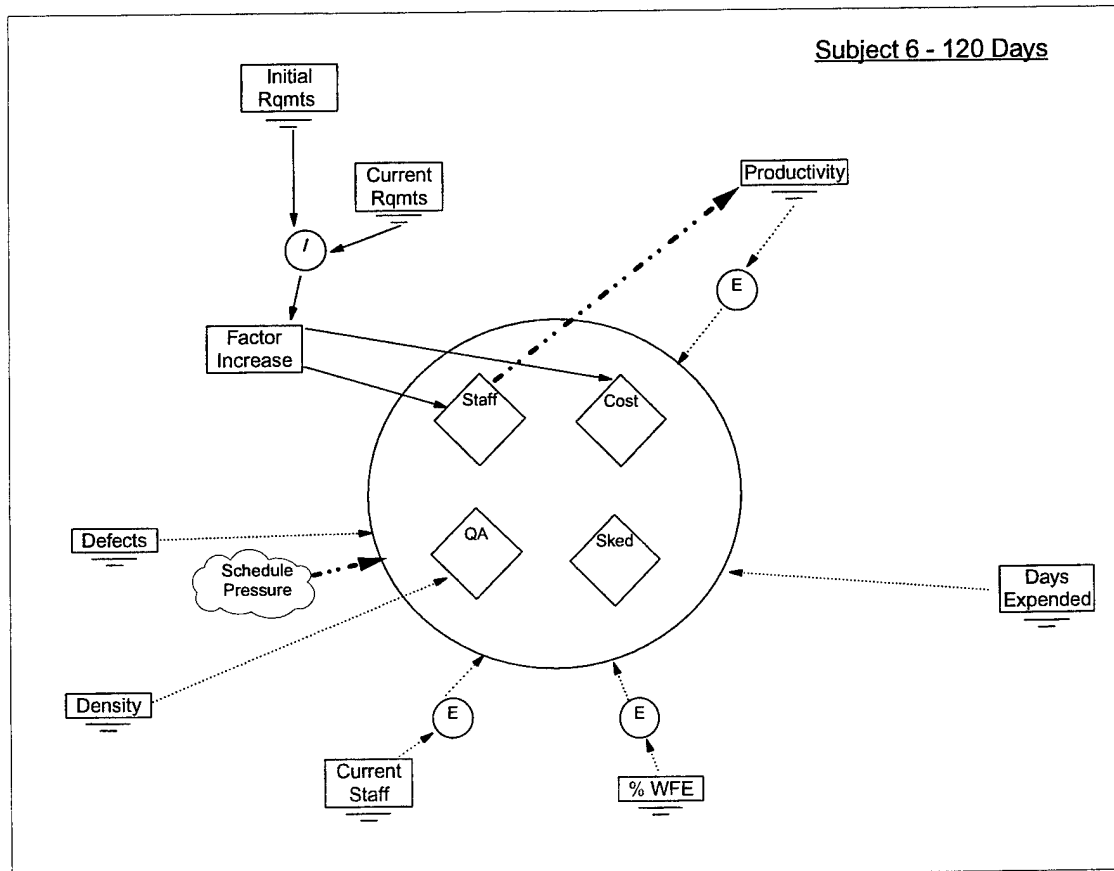
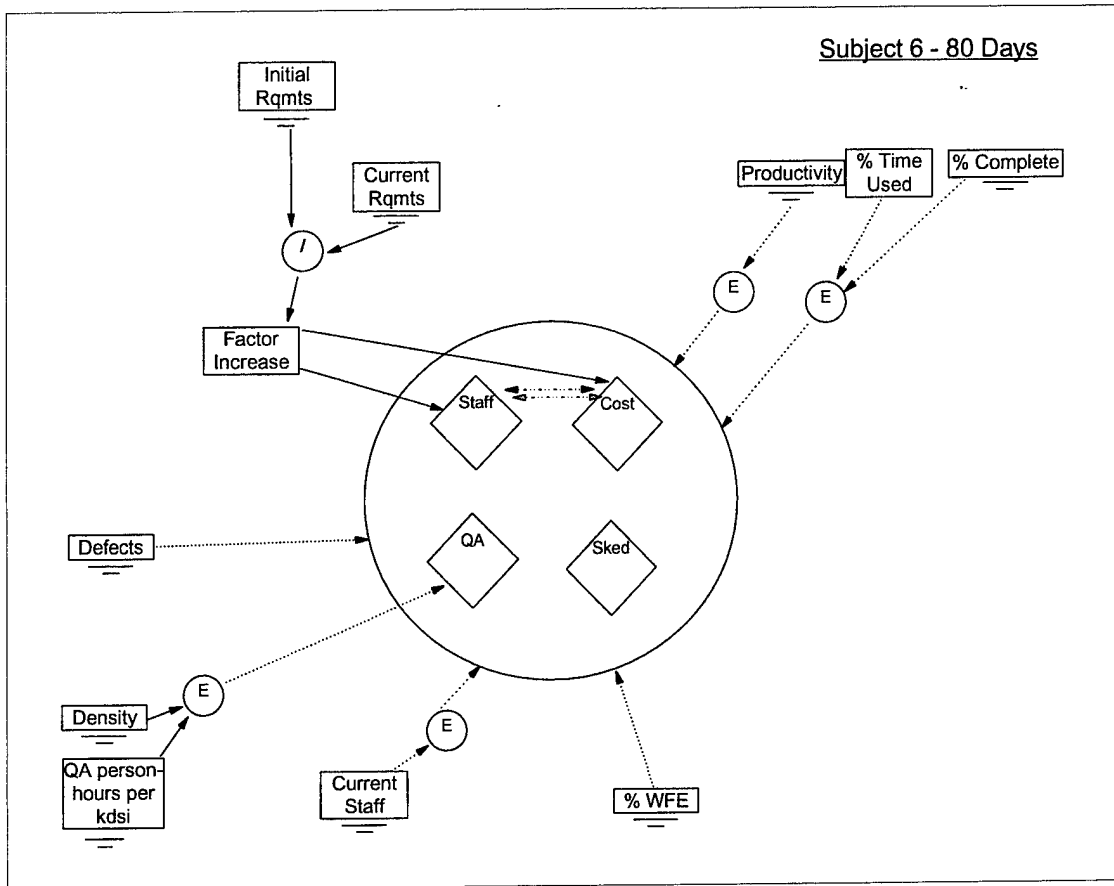


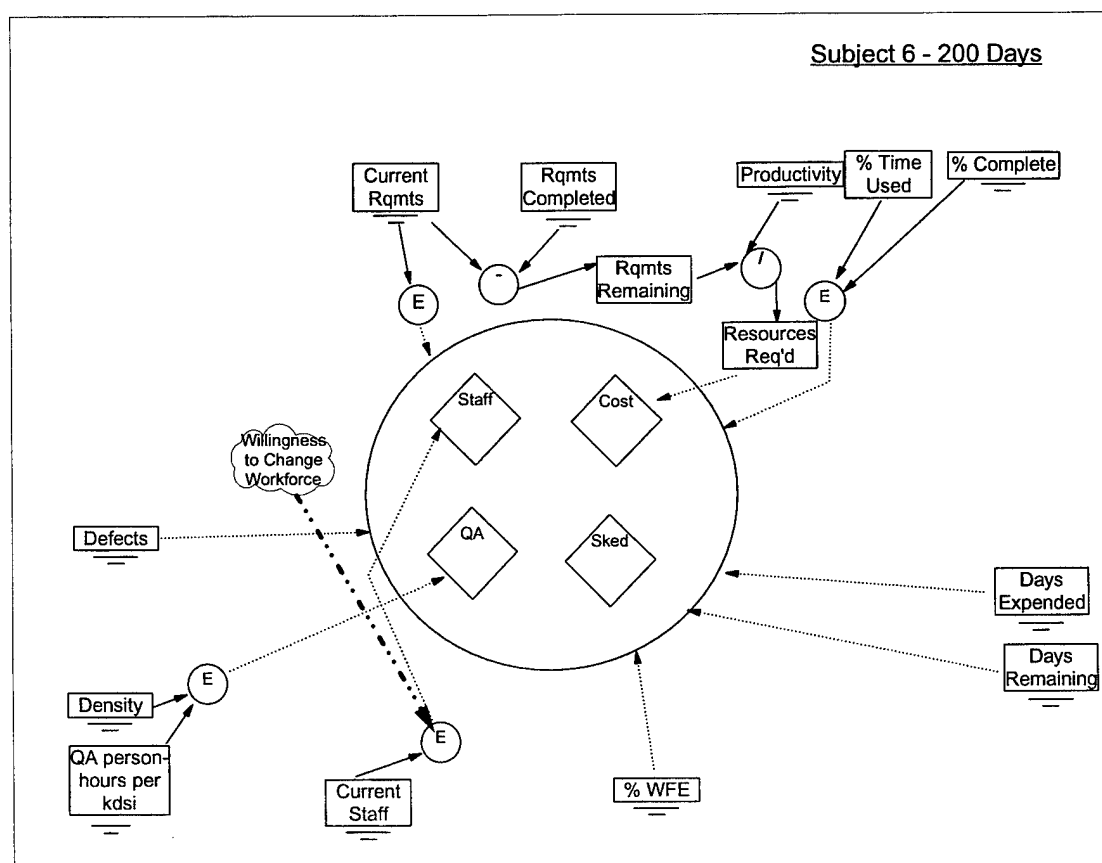
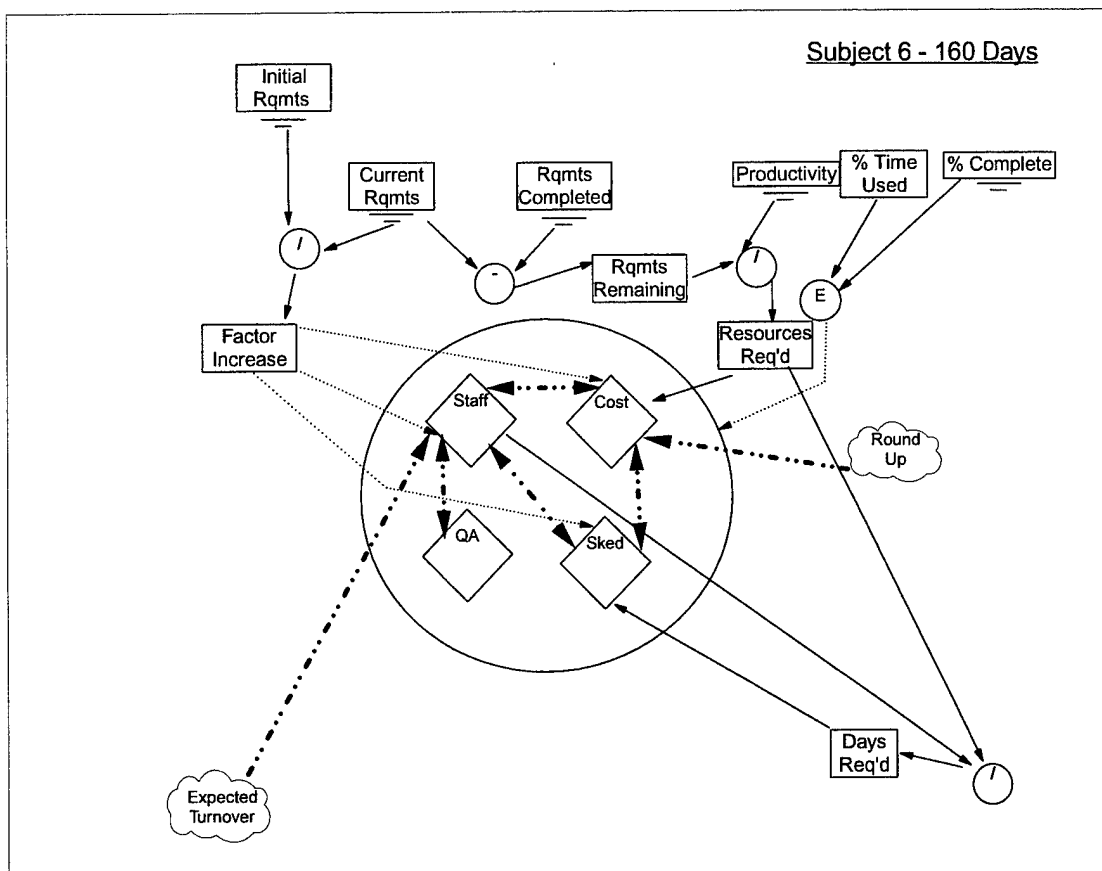










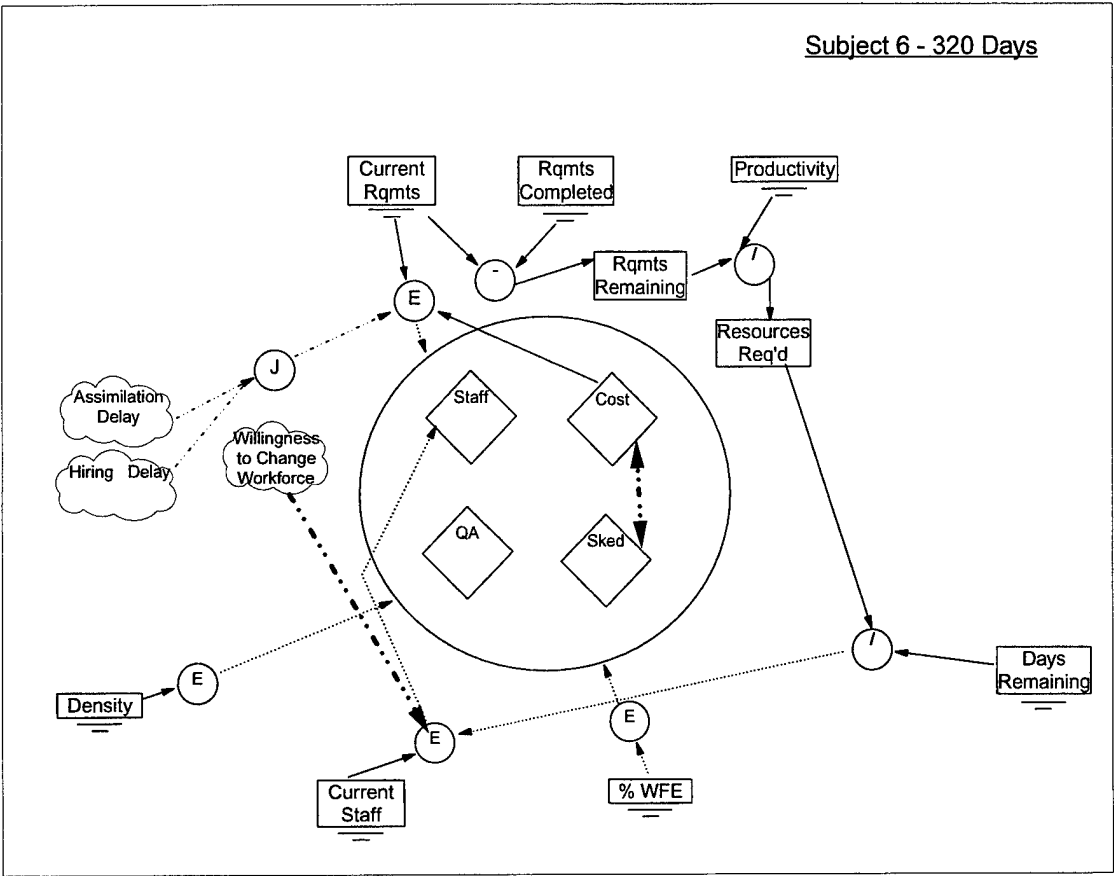


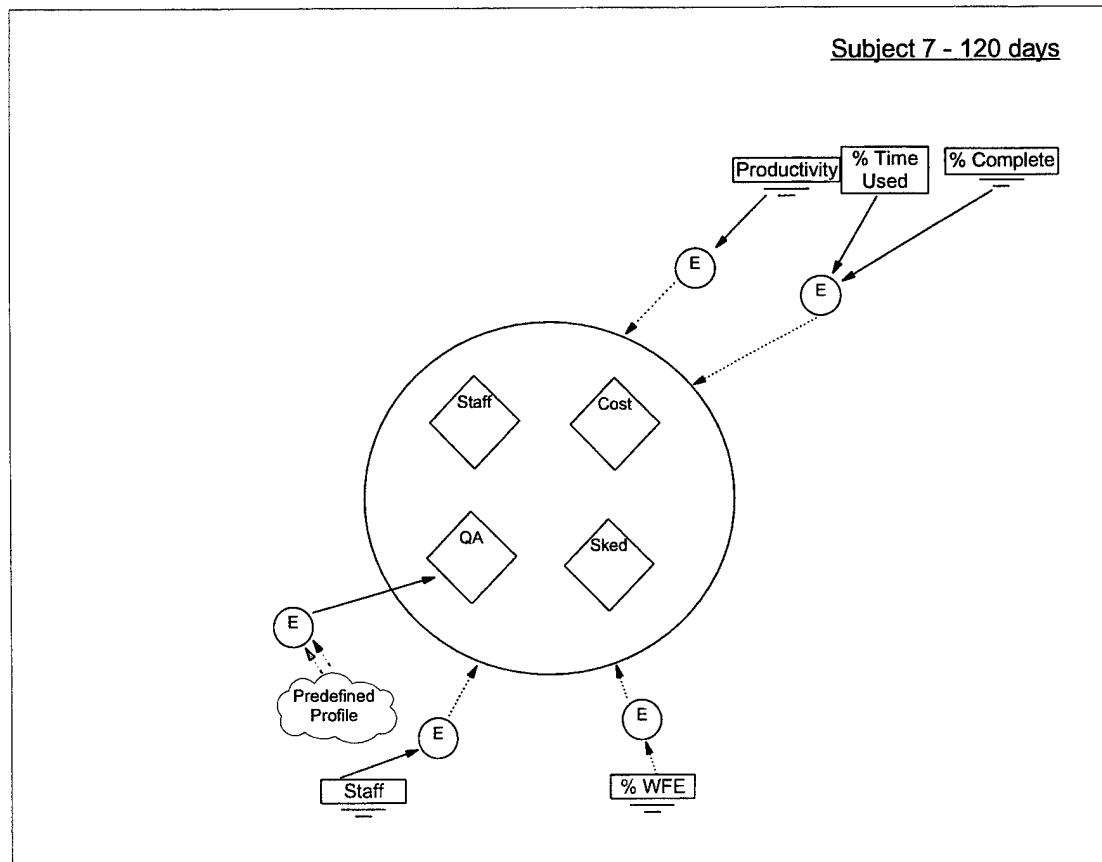
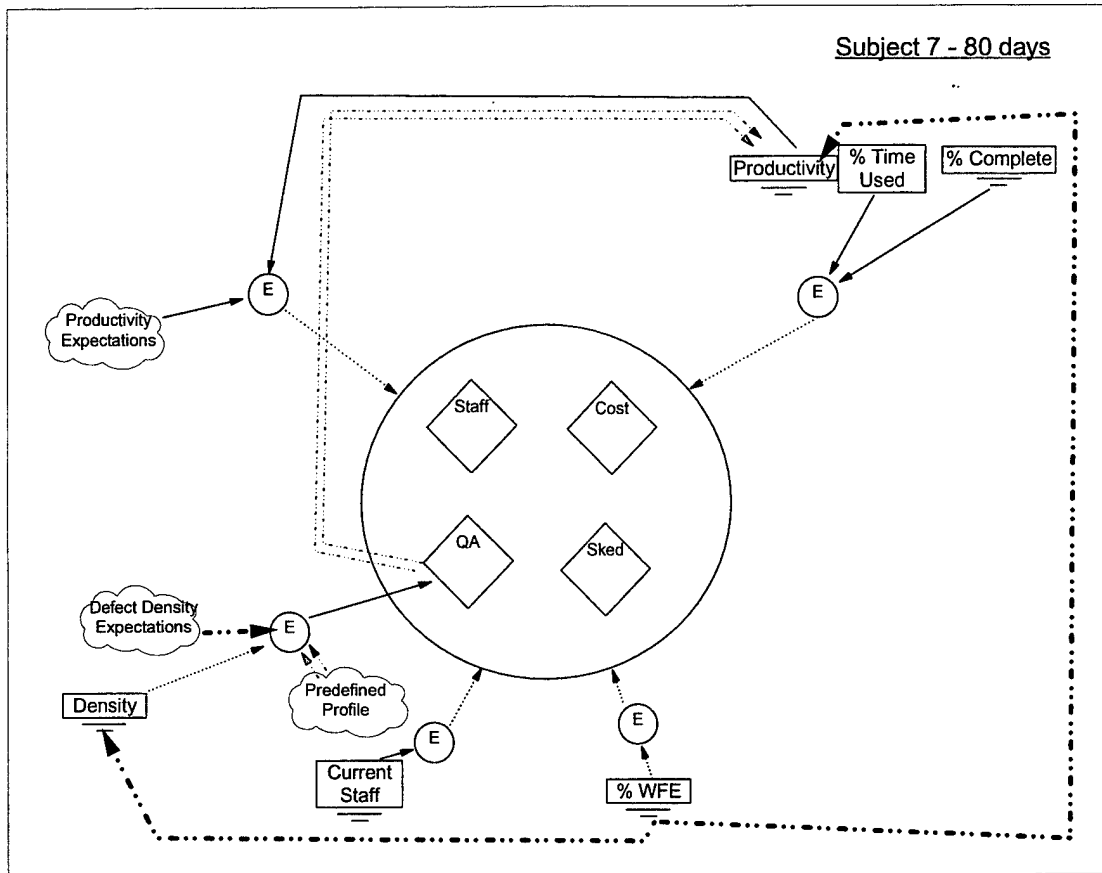
Subject 6 - 240 Days

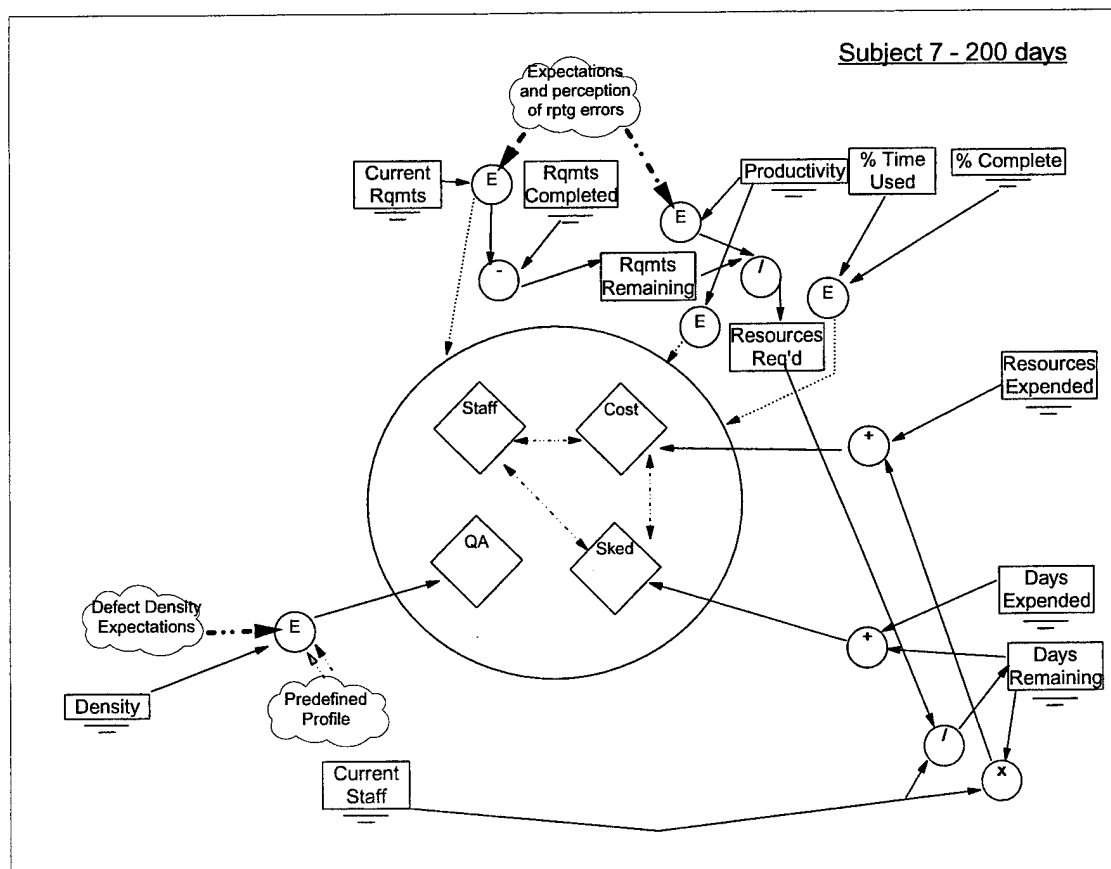
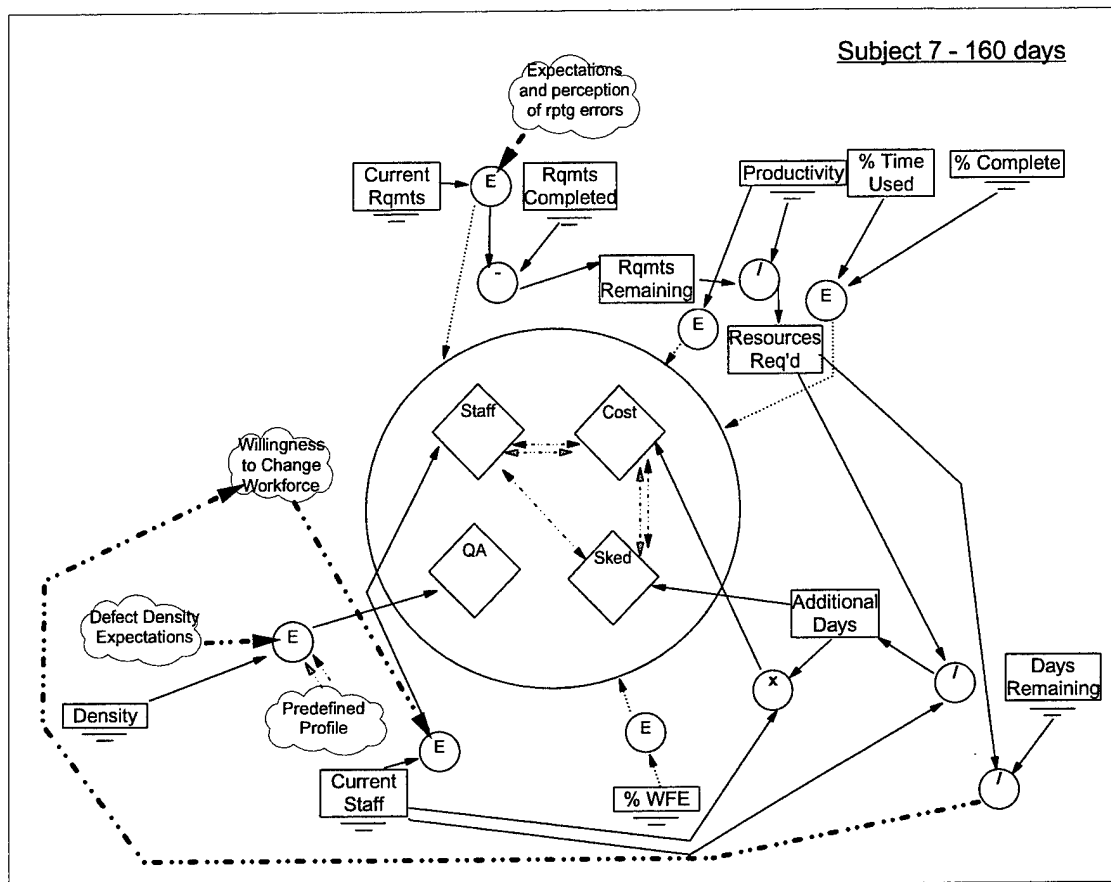
The diagram illustrates a complex system dynamics model for 'Subject 6 - 240 Days'. It features a central circle containing four diamonds representing 'Staff', 'Cost', 'QA', and 'Sked', which are interconnected with each other and with the surrounding system. The model includes several stocks (rectangles) and flows (circles). Key stocks include 'Current Rqmts', 'Rqmts Completed', 'Rqmts Remaining', 'Resources Req'd', 'Days Remaining', 'Days Req'd', 'Current Staff', and '% WFE'. Key flows include 'E' (multiple instances), 'J', and 'I'. A central circle contains four diamonds representing 'Staff', 'Cost', 'QA', and 'Sked', which are interconnected with each other and with the surrounding system. External factors like 'Assimilation Delay', 'Hiring Delay', 'Willingness to Change Workforce', 'Density', and 'QA person-hours per kdsi' are shown as clouds and boxes influencing the system. The diagram is enclosed in a dashed boundary.

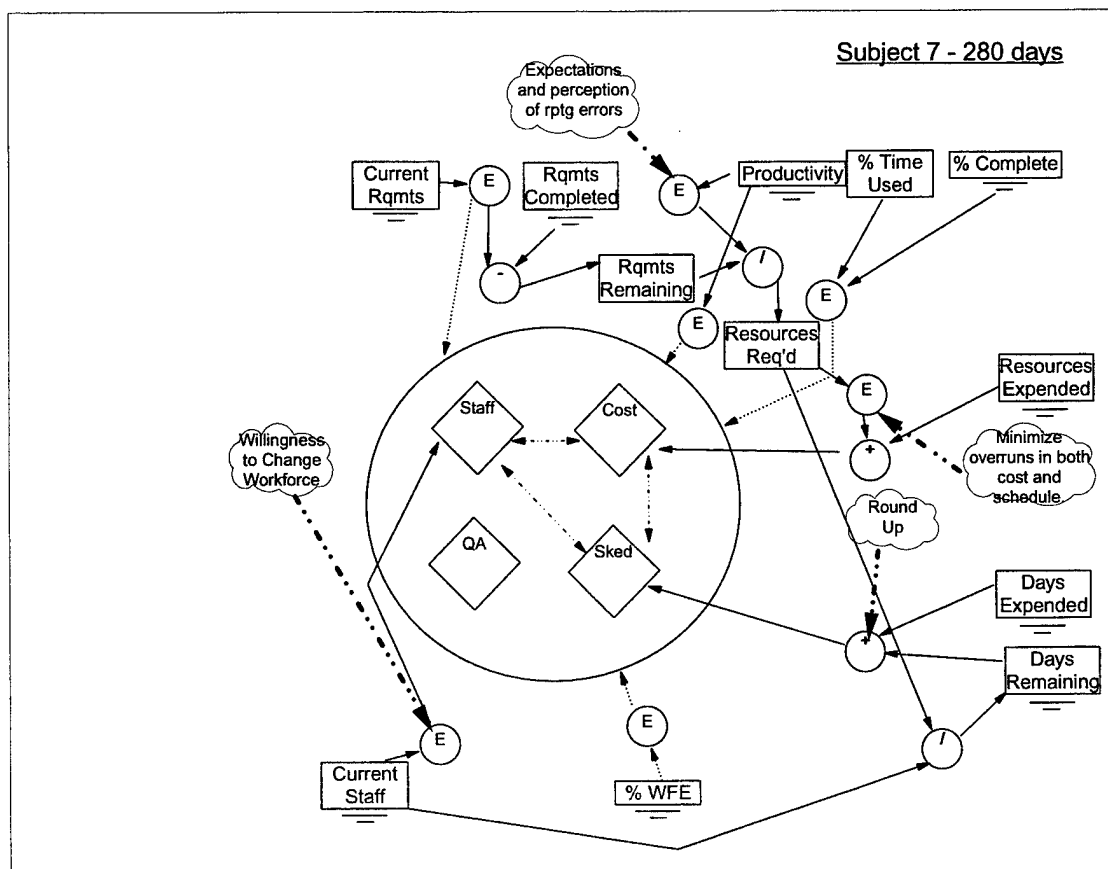
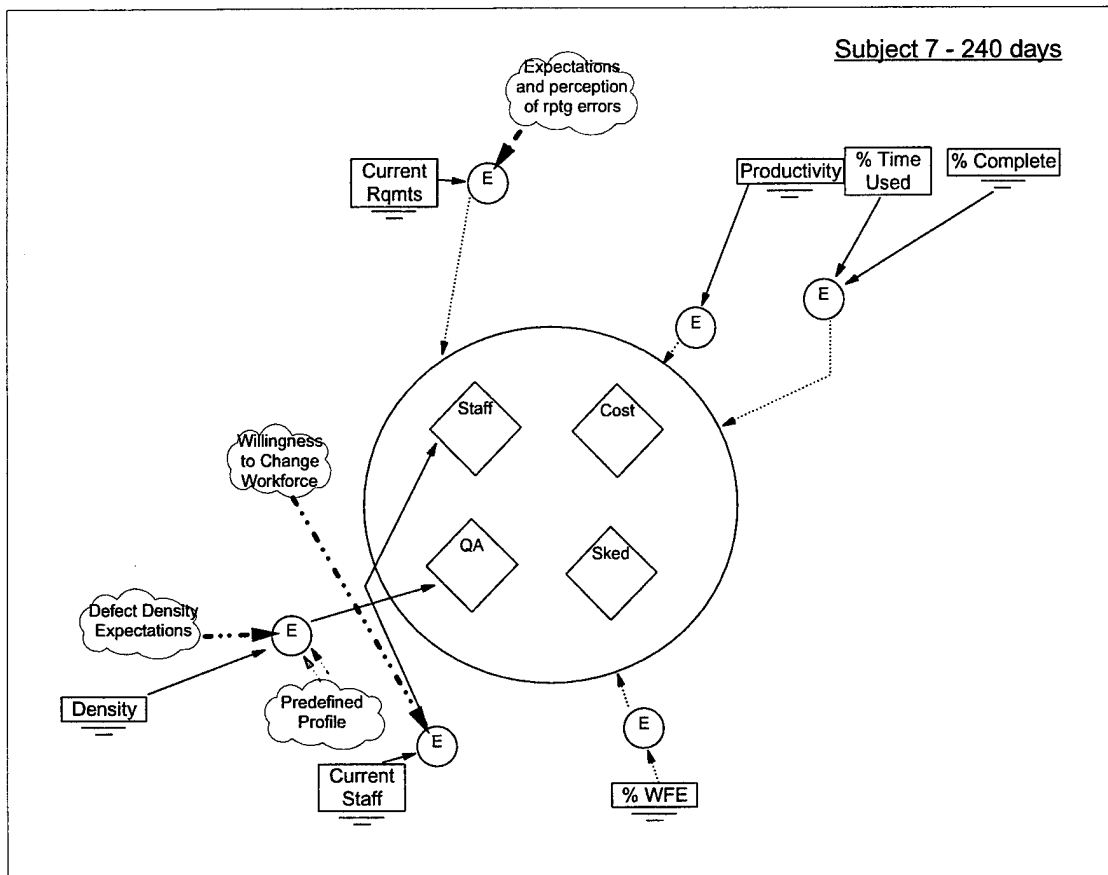
[illegible]

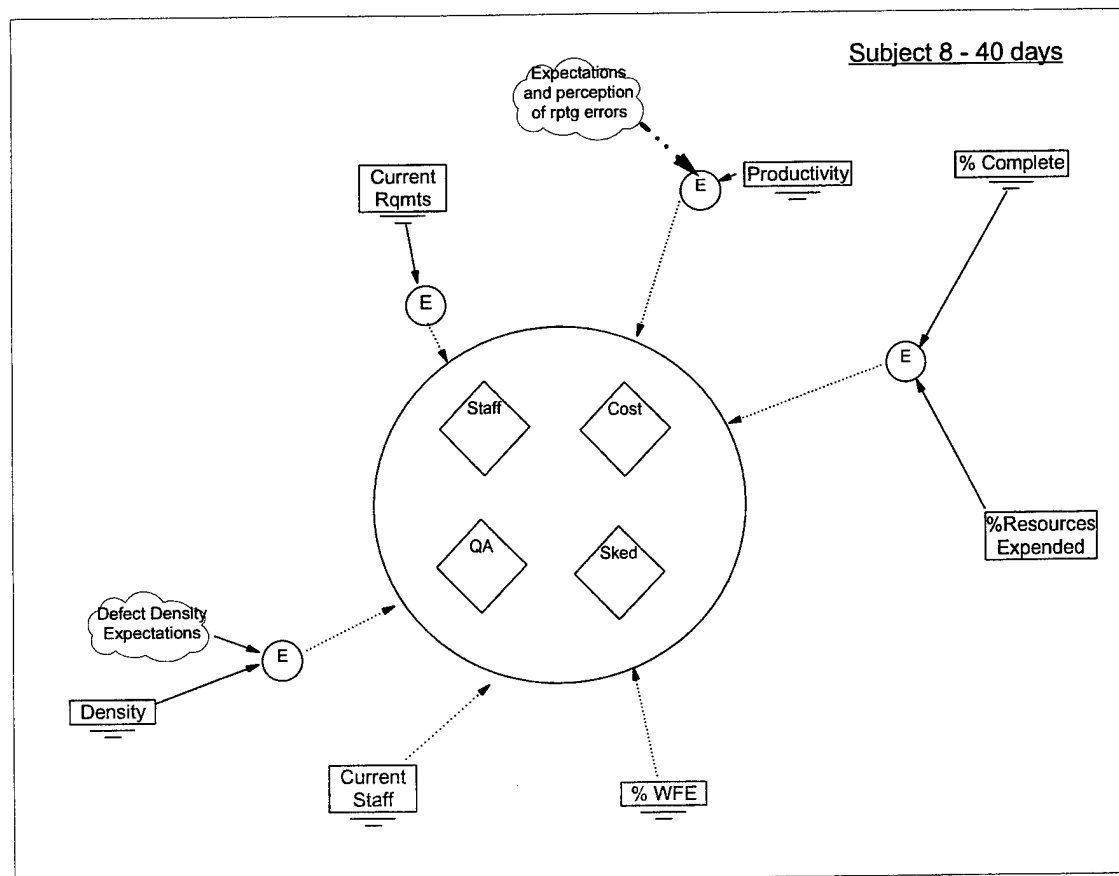
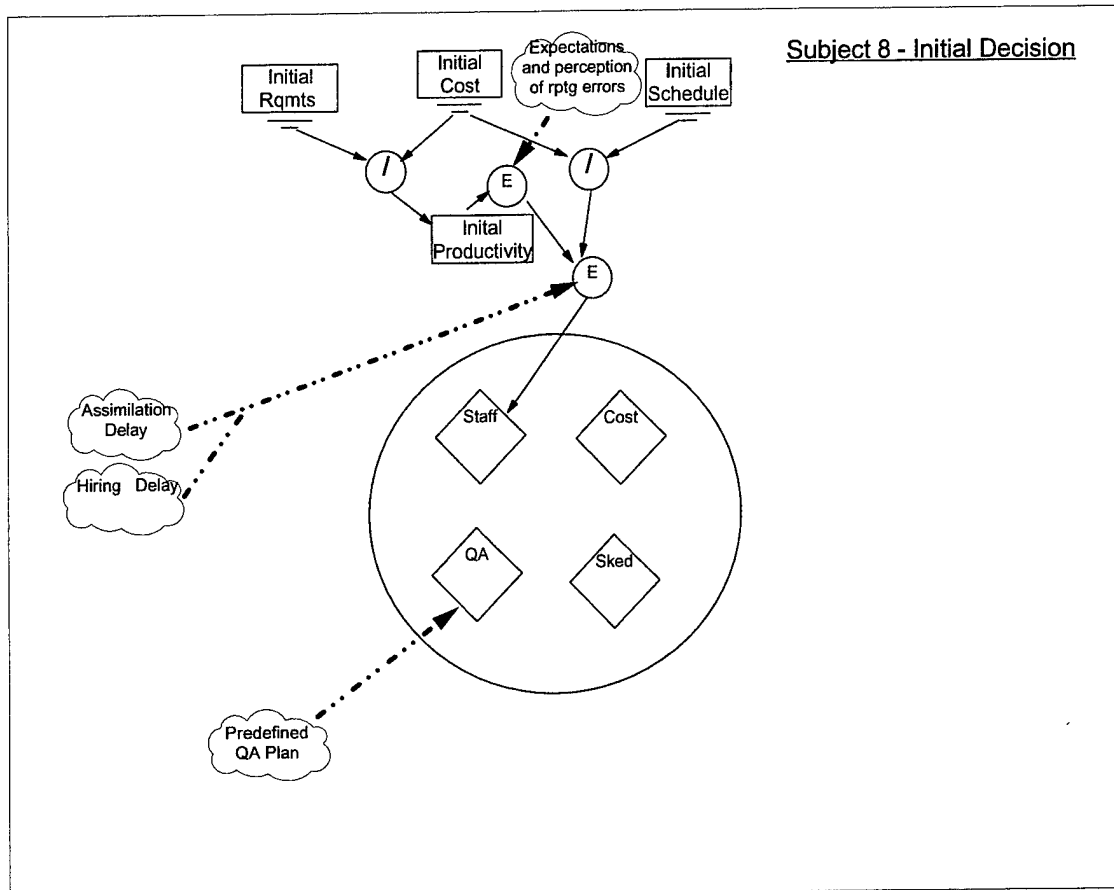
Subject 6 - 320 Days

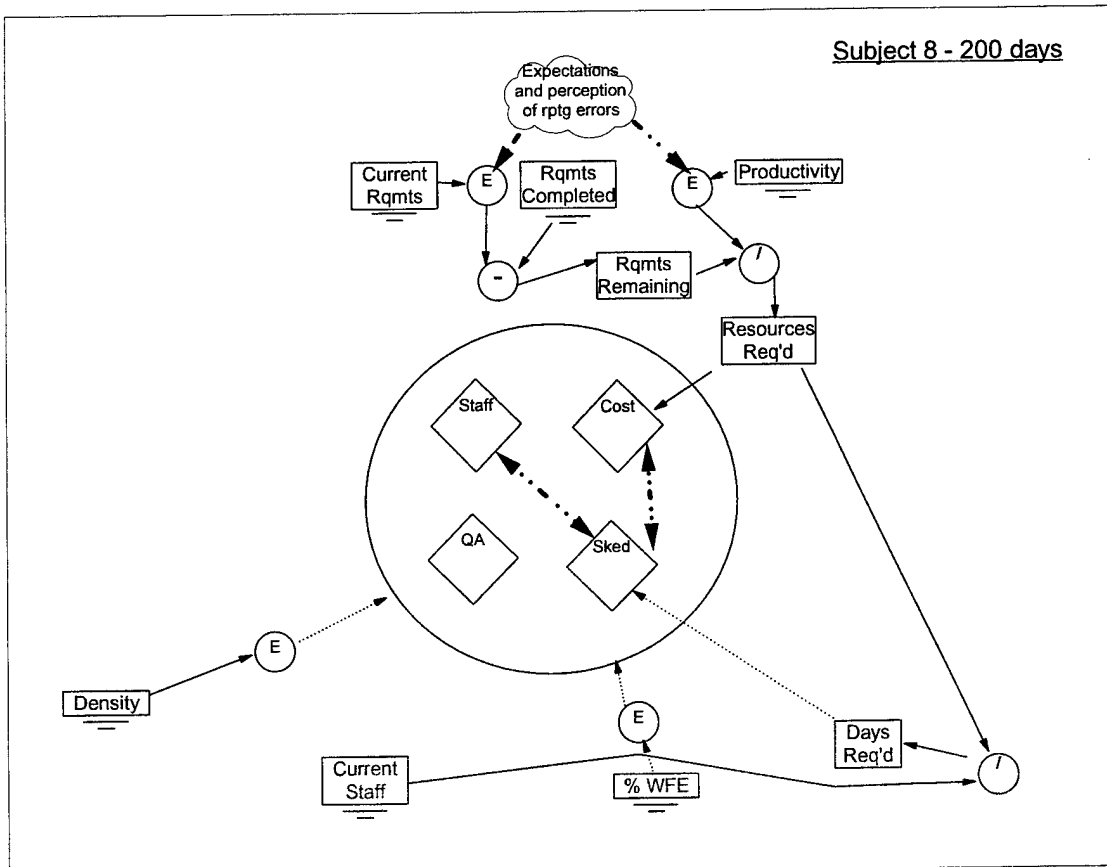
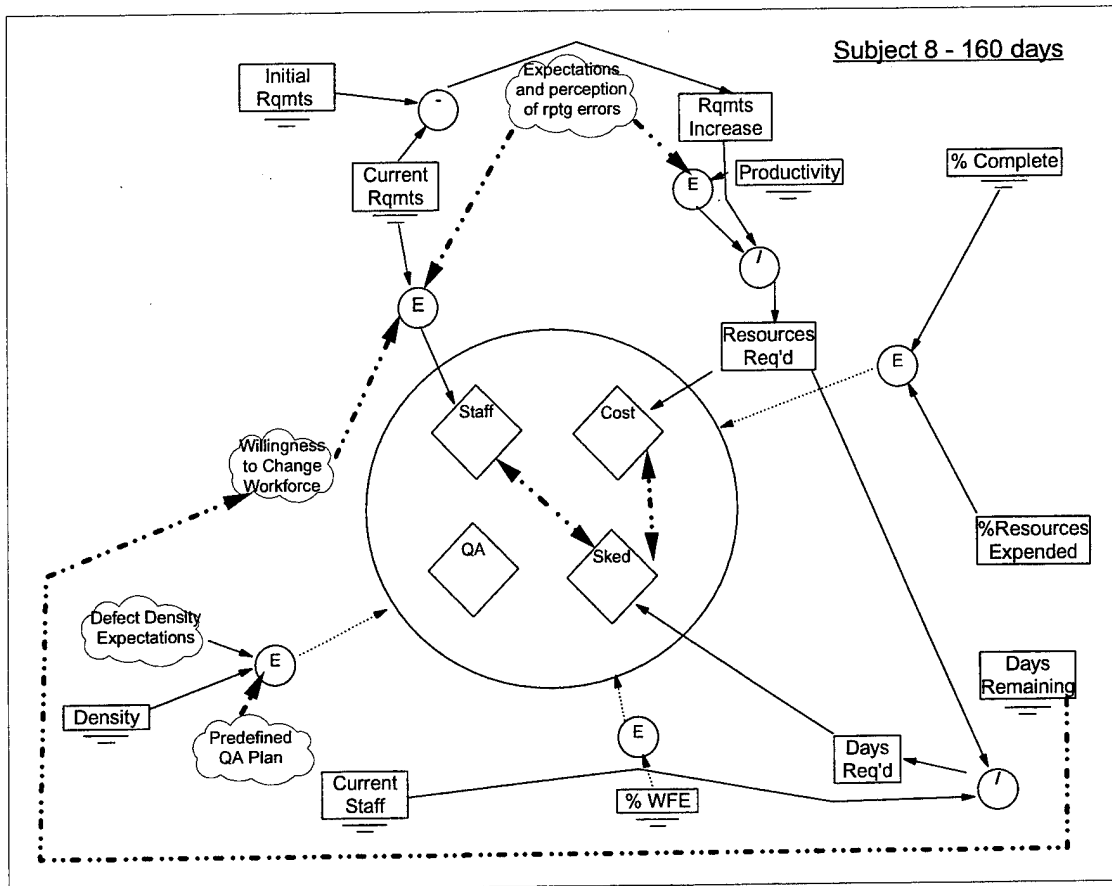


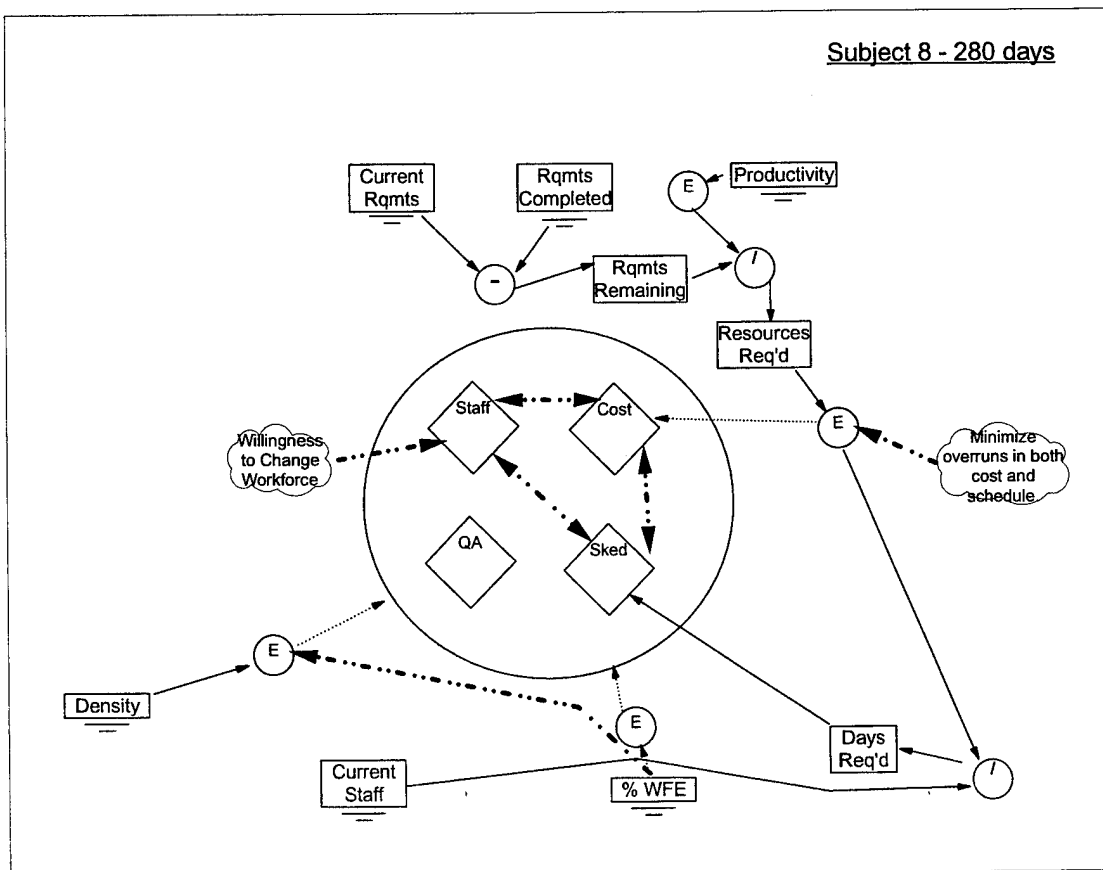
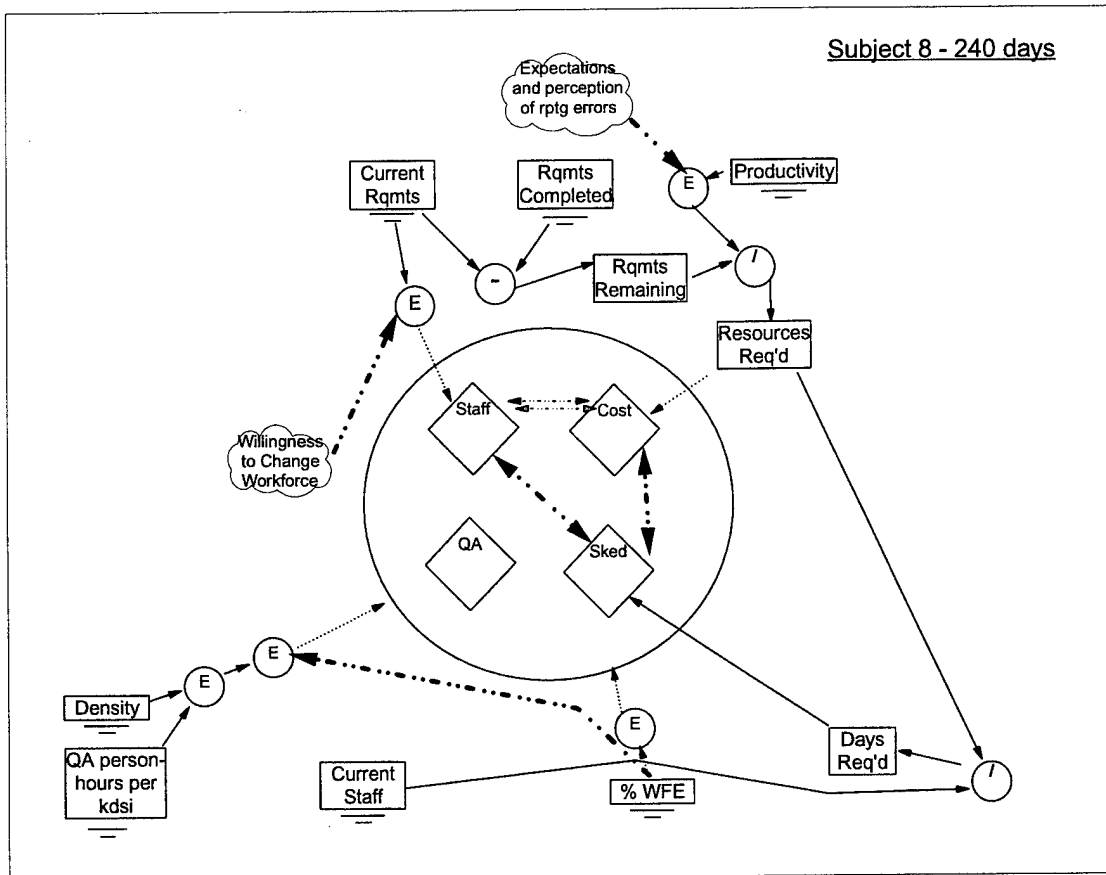


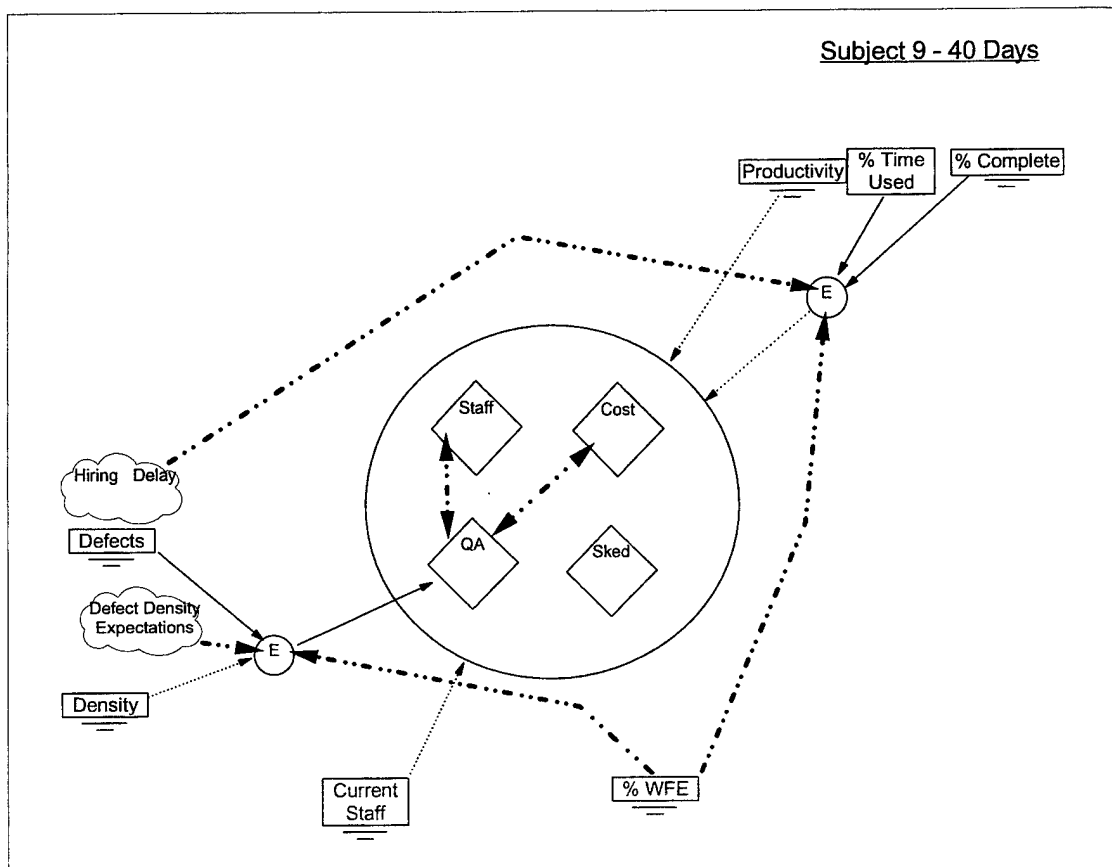
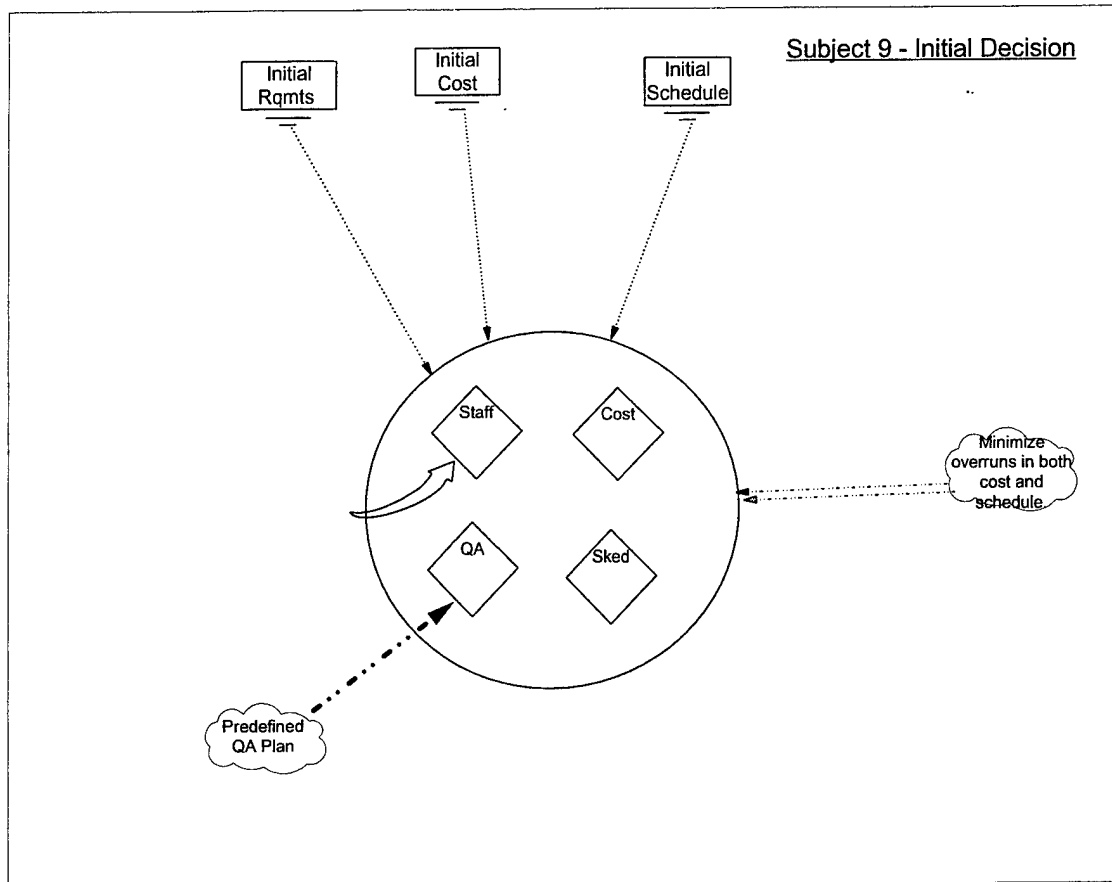


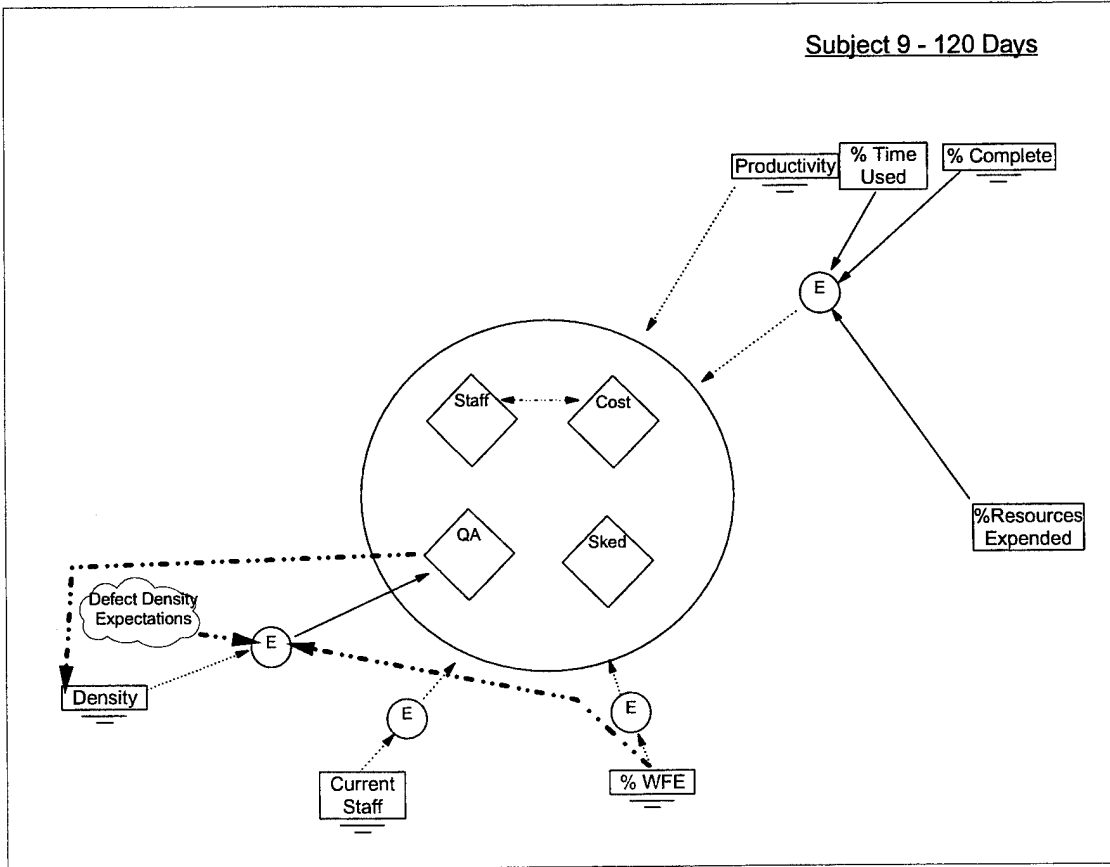
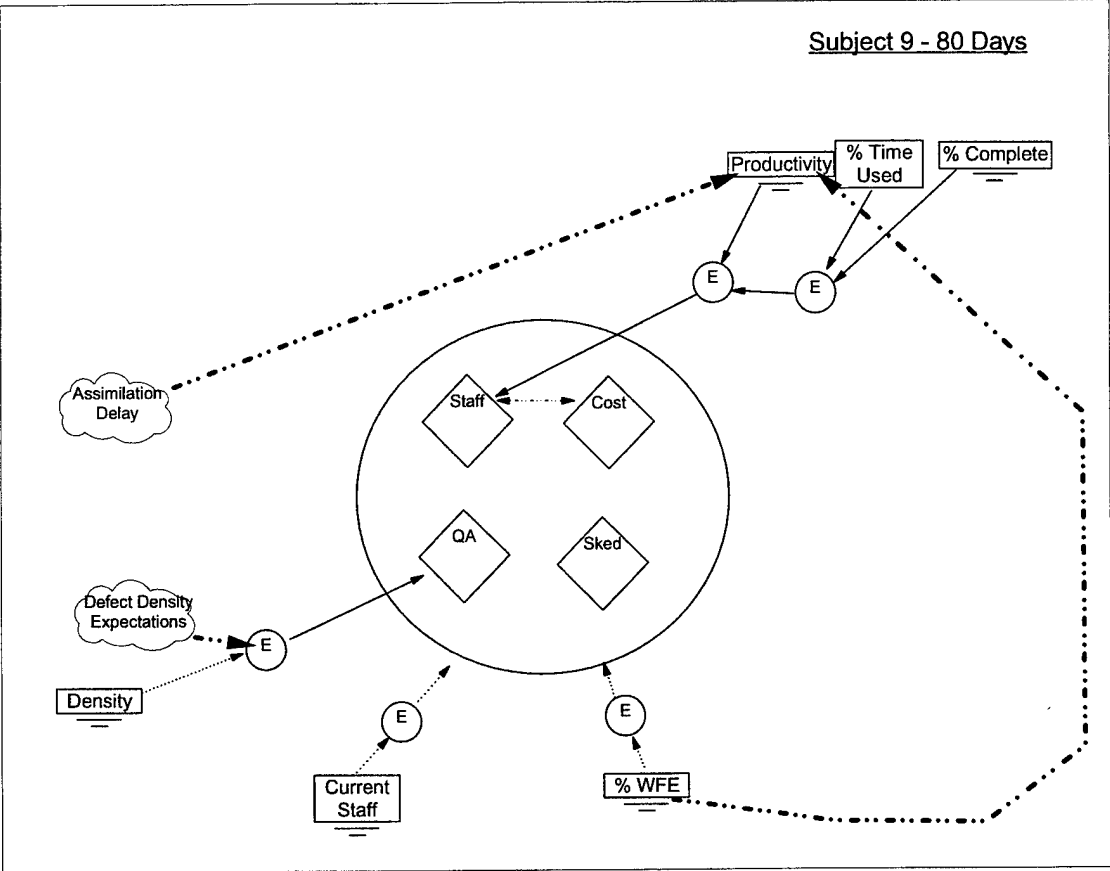


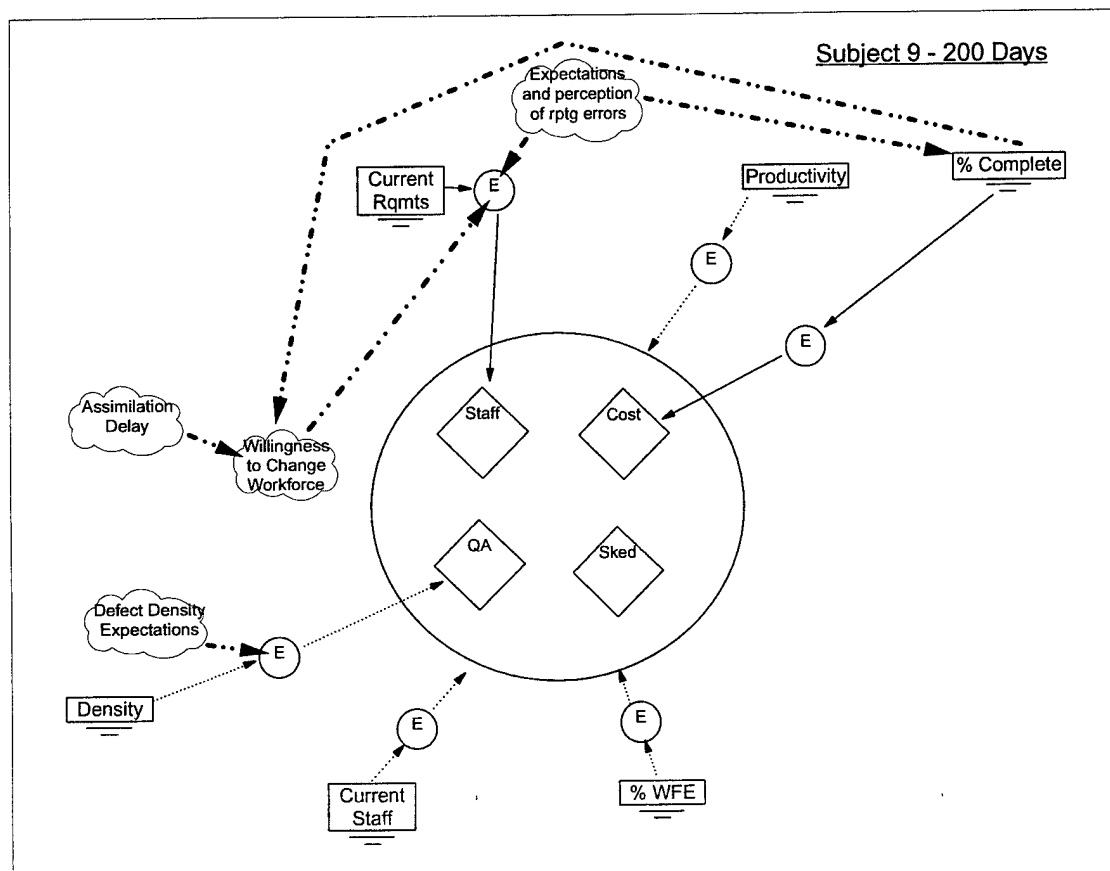
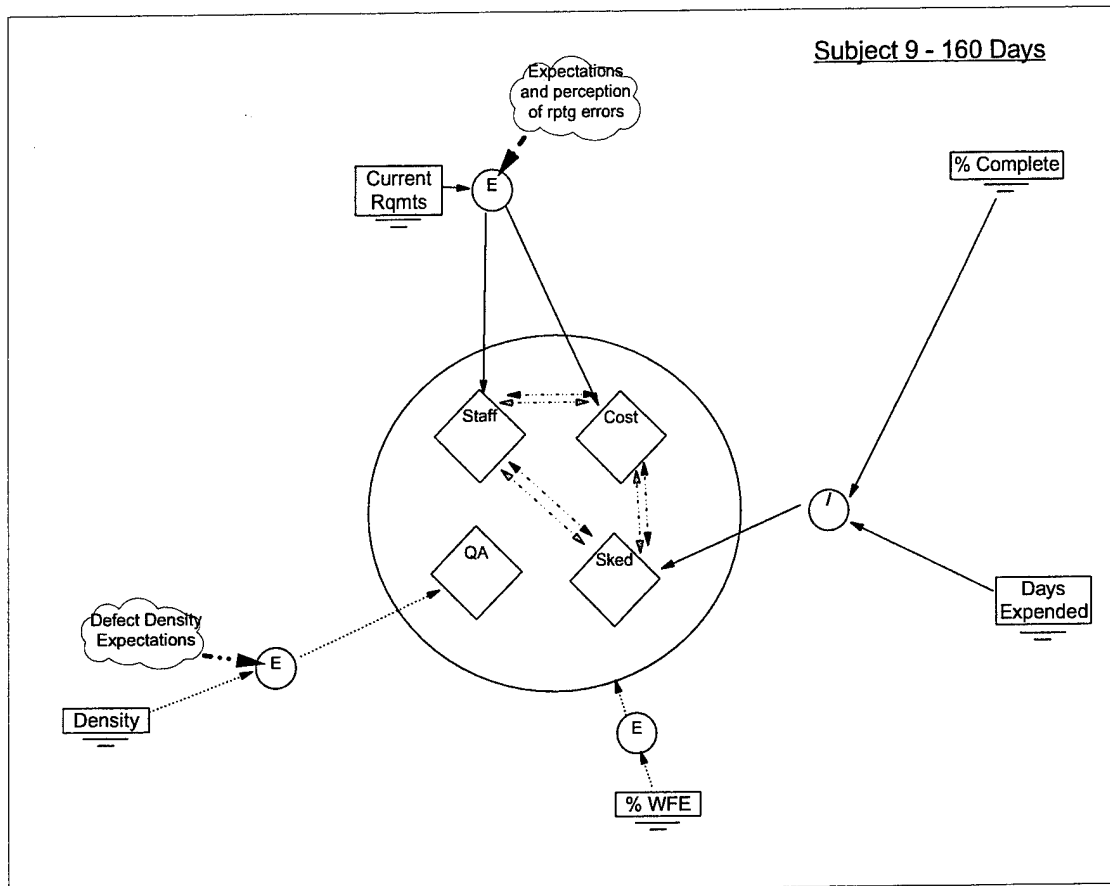


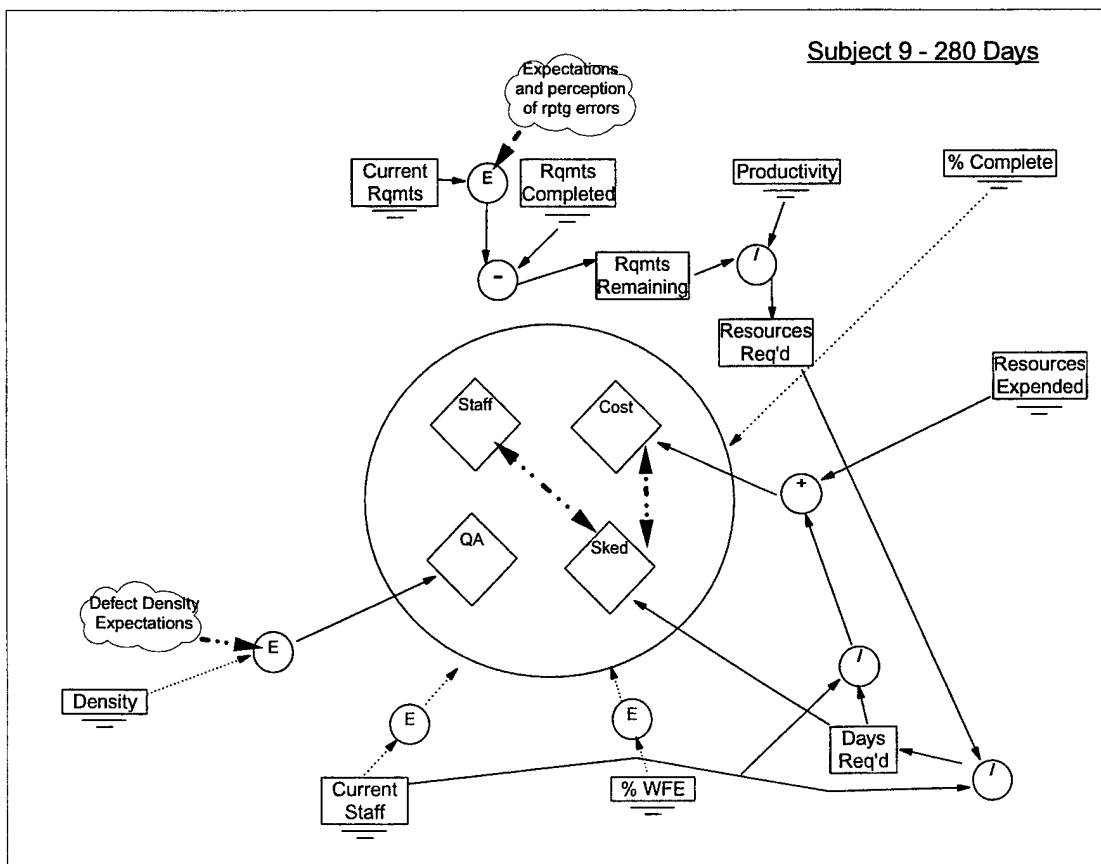
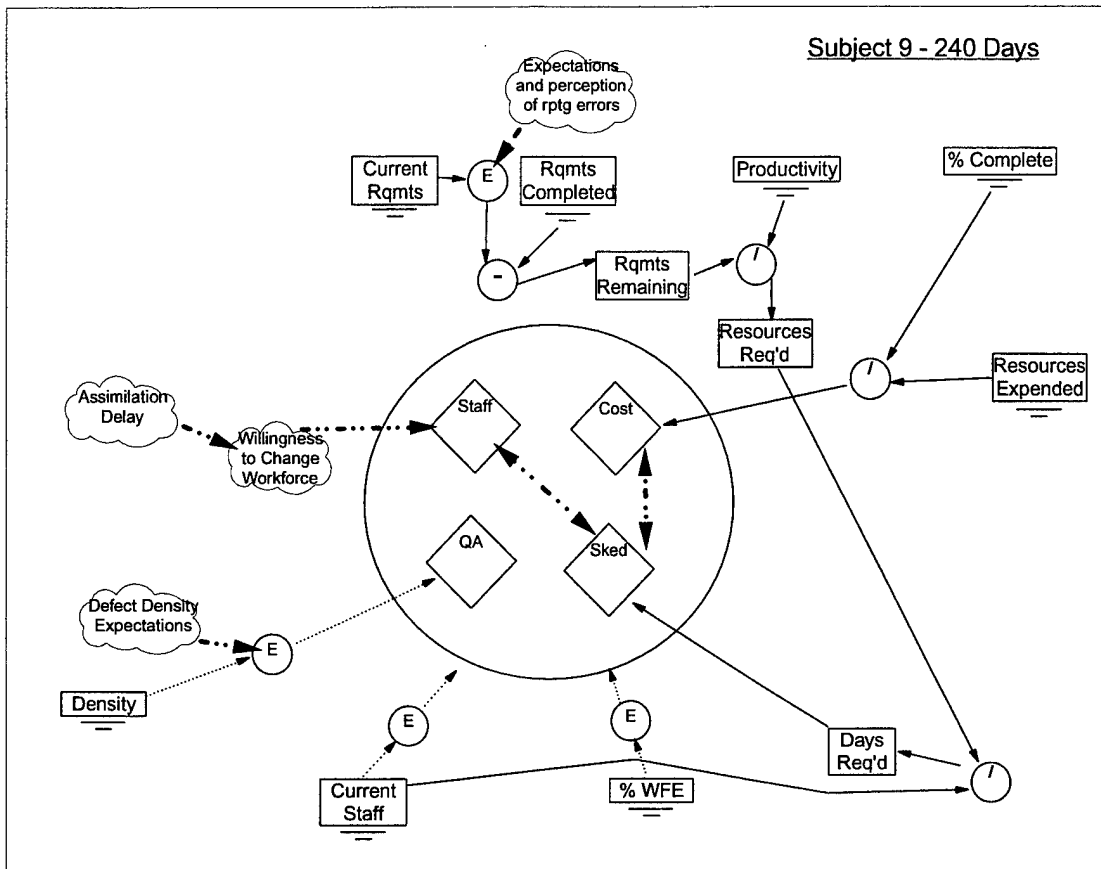


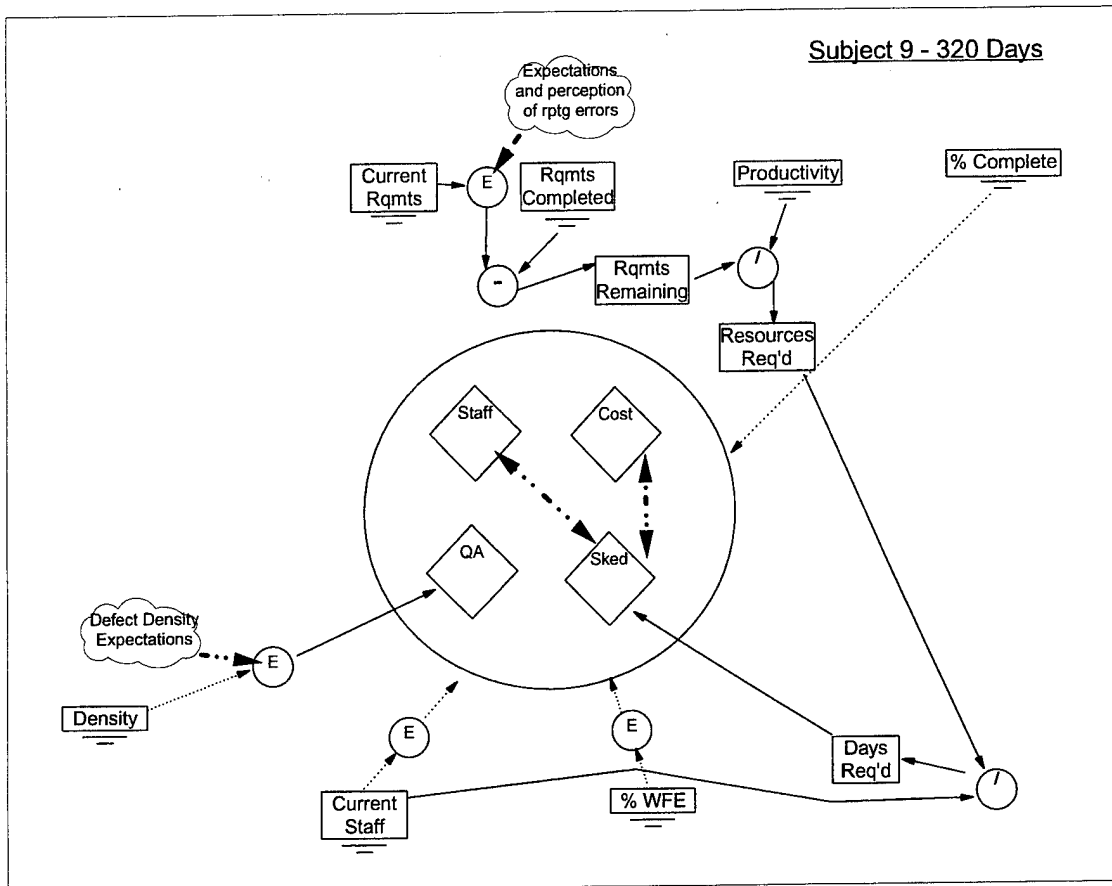




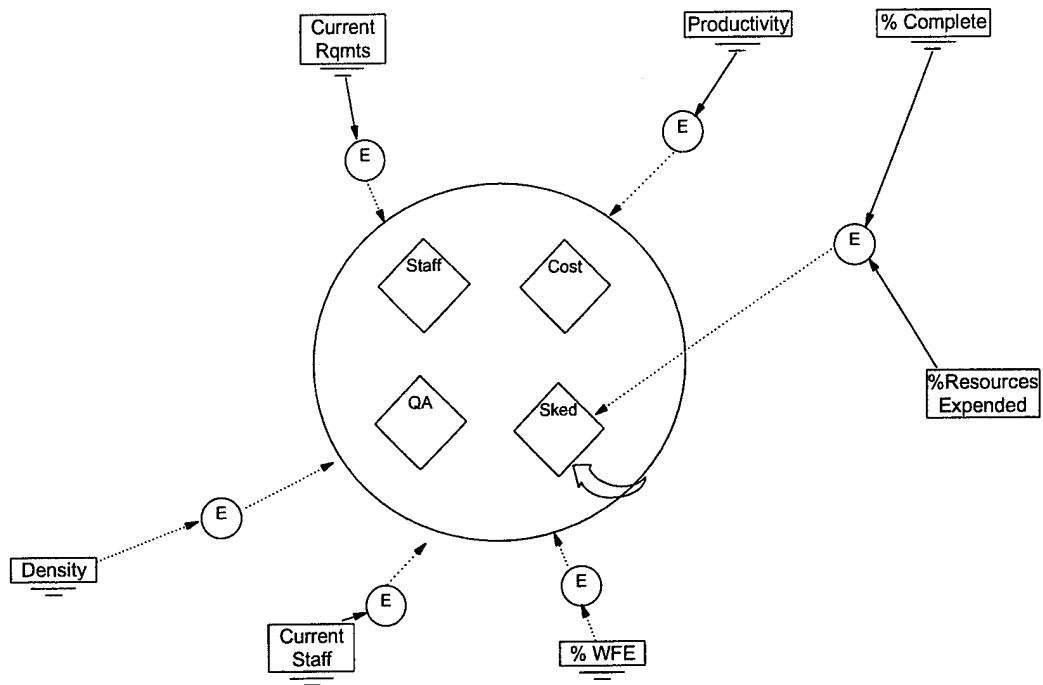




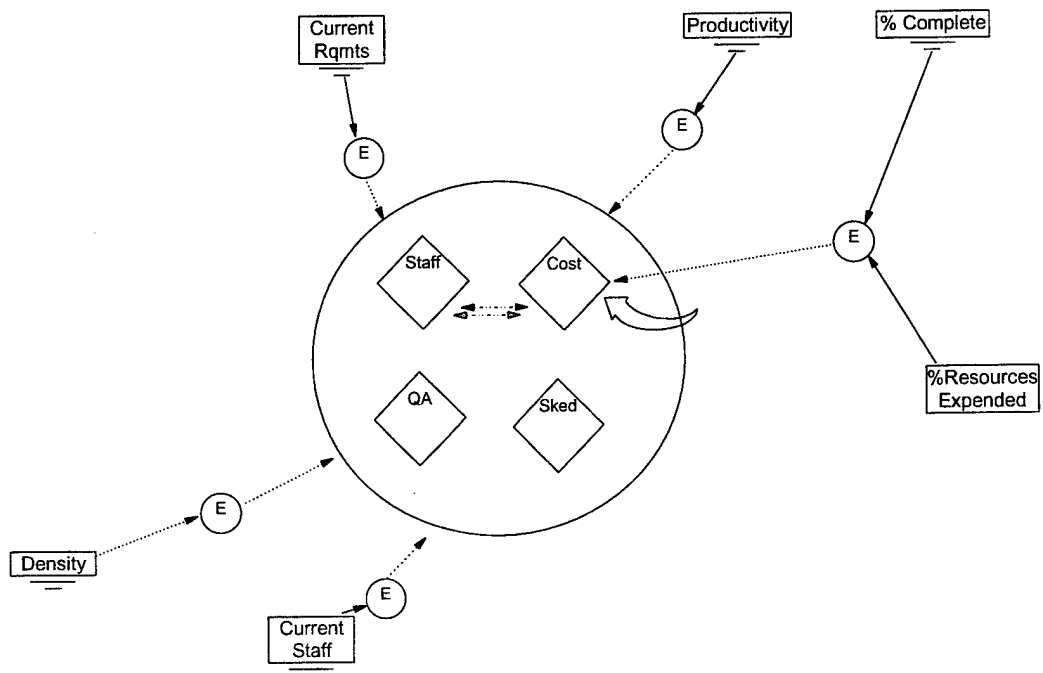




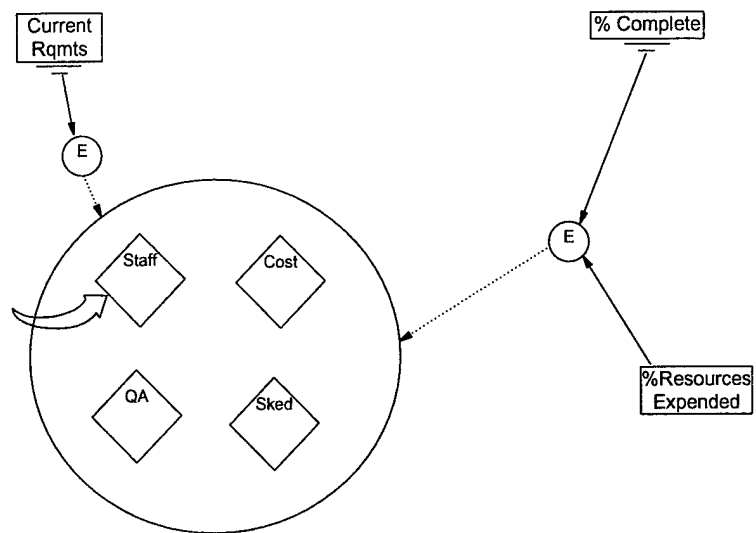
Subject 10 - 80 Days



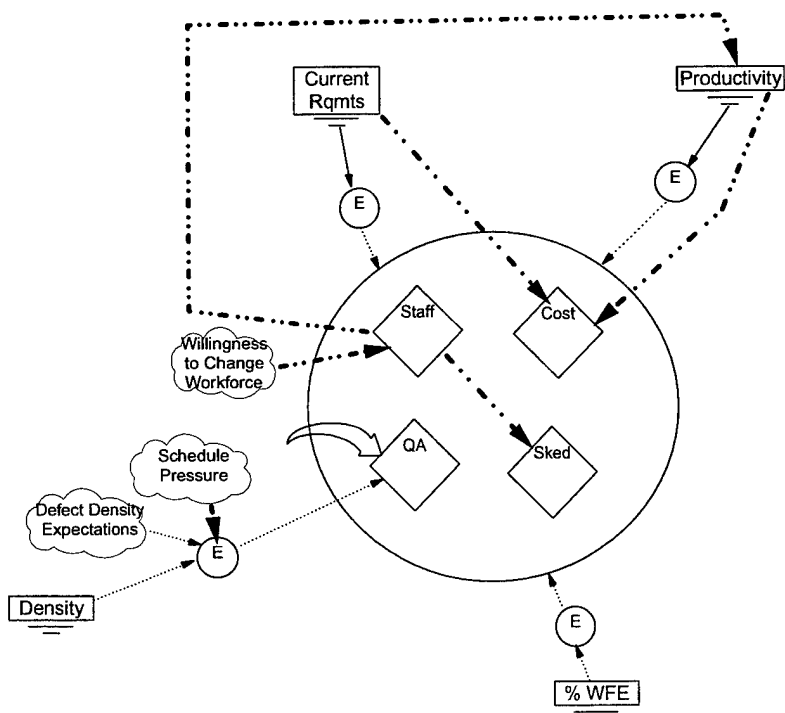
Subject 10 - 120 Days



Subject 10 - 160 Days



Subject 10 - 200 Days



LIST OF REFERENCES

- Abdel-Hamid, T. K., "Investigating the Impacts of Managerial Turnover/Succession on Software Project Performance", *Journal of Management Information Systems*, vol. 9, no. 2, pp. 127-44, Fall 1992.
- Abdel-Hamid, T. K., Madnick, S. E., "Lessons Learned from Modeling the Dynamics of Software Development", *Communications of the ACM*, vol. 32, no. 12, pp. 1426-38, Dec., 1989.
- Abdel-Hamid, T. K., Madnick, S. E., *The Dynamics of Software Development - An Integrative Perspective of the Dynamic Interactions in Software Project Management*, Prentice-Hall, Inc. Englewood Cliffs, N. J., 1990.
- Abdel-Hamid, T. K., Sengupta, K., Ronan, D., "Software Project Control: An Experimental Investigation of Judgment with Fallible Information", *IEEE Transactions on Software Engineering*, vol. 19, no. 6, pp. 603-612, Jun., 1993.
- Adelman, L., Gualtieri, J., Stanford, S., "Examining the Effect of Causal Focus on the Option Generation Process: An Experiment Using Protocol Analysis", *Organizational Behavior and Human Decision Making*, vol. 61, no. 6, pp. 54-66, 1995.
- Baker, D. L., *An Experimental Investigation of the Effects of Software Size increase on Software Project Management Behavior*, M. S. Thesis, Naval Postgraduate School, Monterey, California, Mar., 1992.
- Bosley, M. J., *An Experimental Investigation of Time Delays in Software Project Staffing*, M. S. Thesis, Naval Postgraduate School, Monterey, California, Jun., 1994.
- Brooks, F. P. Jr., *The Mythical Man-Month*, Reading, Addison-Wesley Publishing, MA, 1975.
- Camerer, C. F., Johnson, E. J., "The Process-Performance Paradox in Expert Judgment", *Toward a General Theory of Expertise*, edited by Ericsson and Smith, Chapter 8, Cambridge University Press, Cambridge, 1991.
- Carroll, J. S., Johnson, E. J., *Decision Research - A Field Guide*, Sage Publications, Newbury Park, California, 1990.
- Ericsson, K. A., Simon, H. A., *Protocol Analysis - Verbal Reports as Data*, The Massachusetts Institute of Technology, Cambridge, Massachusetts, 1984.
- Ford, J. K., Schmitt, N., Schechtman, S. L., Hults, B. M., Doherty, M. L., "Process Tracing Methods: Contributions, Problems, and Neglected Research Questions", *Organizational Behavior and Human Decision Processes*, no. 43, pp. 75-117, 1989.
- Funke, J., "Solving Complex Problems: Exploration and Control of Complex Systems", Chapter 6 of *Complex Problem Solving Principles and Mechanisms*, edited by Sternberg and French, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1991.
- GAO, *Automated Information Systems - Schedule Delays and Cost Overruns Plague DoD Systems*, Report IMTEC-89-36, May 10, 1989.

GAO, *DoD's CAL Initiative*, Report AIMD-94-197R, September 30, 1994.

Hardebeck, M. J., *Decision Making for Software Project Management in a Multi-project Environment: An Experimental Investigation*, M.S. Thesis, Naval Postgraduate School, Monterey, California, September, 1991.

Kitfield, J., "Is Software DoD's Achilles Heel?", *Military Forum*, July 1989.

Schweiger, D. M., Anderson, C. R., Locke, E. A., "Complex Decision Making: A Longitudinal Study of Process and Performance", *Organizational Behavior and Human Decision Making Processes*, vol. 36, pp. 245-272, 1985.

Sterman, J. D., *Learning in and About Complex Systems*, Sloan School of Management, Massachusetts Institute of Technology, 1994.

Svenson, O., "Eliciting and Analyzing Verbal Protocols in Process Studies of Judgment and Decision Making", *Process and Structure in Human Decision Making*, edited by Montgomery and Svenson, Chapter 4, John Wiley and Sons Ltd., New York, New York, 1989.

Swett, C. L., *An Experimental Investigation of the Impact of Conflicting Project Goals on Staff Resource Allocation*, M. S. Thesis, Naval Postgraduate School, Monterey, California, Mar., 1992.

BIBLIOGRAPHY

- Brehmer, B., Lovborg, L., and Winman, A., *Learning the Time Constants of a Dynamic Task*, Centre for Cognitive Science, University of Roskilde, WPCS-92-6, November 1992.
- Gibbs, B. J., Hogarth, R. M., Marquis, M. A., McKenzie, C. R. M., "Learning from Feedback: Exactingness and Incentives", *Journal of Experimental Psychology: Learning, Memory and Cognition*, vol. 17, no. 4, pp. 734-752, 1991.
- Morecroft, J. D. W., "System Dynamics and Microworlds for Policy Makers", *European Journal of Operational Research*, vol. 35, pp. 301-320, 1988.
- Pennington, N., Lee, A. Y., Rehder, B., "Cognitive Activities and Levels of Abstraction in Procedural and Object-Oriented Design", *Human-Computer Interaction*, vol. 10, pp. 171-226, Lawrence Erlbaum Associates, 1995.
- Roschelle, J., Pea, R., and Trigg, R., *Videonoter: A Tool for Exploratory Video Analysis*, Institute for Research on Learning, IRL Report No. IRL90-0021, March 1990.
- Sengupta, K., Abdel-Hamid, T. K., "Alternative Conceptions of Feedback in a Dynamic Decision Environment", *Management Science*, vol. 39, no. 4, pp. 411-428, Apr., 1993.
- Sengupta, K., Abdel-Hamid, T. K., "The Impact of Unreliable Information on the Management of Software Projects: A Dynamic Perspective", *IEEE Transactions on Systems, Man, and Cybernetics*, Forthcoming.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center2
Cameron Station
Alexandria, Virginia 22304-6145
2. Library, Code 522
Naval Postgraduate School
Monterey, California 93943-5101
3. Dr. Tarek Abdel-Hamid2
Department of Systems Management
Code SM/Ah
Naval Postgraduate School
Monterey California 93943-5002
4. Dr. Kishore Sengupta2
Department of Systems Management
Code SM/Se
Naval Postgraduate School
Monterey, California 93943-5002
5. Dr. Balasubranianiam Ramesh2
Department of Systems Management
Code SM/Ra
Naval Postgraduate School
Monterey, California 93943-5002
6. LCDR Douglas W. Swanson.....2
131 Silverwood
Slidell, Louisiana 70461